UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

Doctoral Dissertation
Doctoral Program in Mathematics and its Applications (34^{th}cycle)

Physics-Informed Numerical Analysis and Machine Learning for Modelling and Analysis of Complex Systems with Applications in Computational Neuroscience

By

Galaris Evangelos

******

Supervisor:
Prof. Siettos Constantinos

University of Naples Federico II

2022

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Galaris Evangelos
2022

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Department of Mathematics and its Applications "Renato Caccioppoli" of University of Naples Federico II (UNINA).

I would like to dedicate this thesis to the non-privileged. To every person that gives its internal and external conflicts.

"Il mondo é un bel libro, ma poco serve a chi non lo sa leggere
(The world is a beautiful book, but of little use to him who cannot read it.)".
(Carlo Goldoni, "La Pamela")

Acknowledgements

The present thesis has been carried out at the Department of Mathematics and Applications "Renato Caccioppoli" of the University of Naples "Federico II" under the valuable supervision of Prof. Constantinos Siettos. I would like to thank him deeply for his selfless support during my Ph.D. program.

I am also grateful to the University of Naples "Federico II" for the three year scholarship that supported my research work. During my Ph.D. period I had the opportunity to meet great persons: colleagues, professors, university staff and friends. I enjoyed every time in the university and in the city of Naples and I am full of great memories.

Finally, I would like to thank my parents Dimitrios and Alexandra, my brothers Apostolos and Pierros and some special friends for being by my side during good and bad periods. Without their presence, I would never be arrived in that moment.

Abstract

This Ph.D. Thesis deals with the development and implementation of physics-informed numerical analysis and machine learning methods for the modelling and bifurcation analysis of large-scale complex dynamical systems from big-data as well as the numerical solution of low-dimensional models of Ordinary Differential Equations (ODEs). Modelling and analysing the emergent behaviour of large-scale complex dynamical systems from experimental data and/or data produced by detailed high-fidelity microscopic simulations, requires appropriate data-driven/numerical analysis-based methods for extracting coarse-scale models that can be utilized for further numerical analysis of the emergent dynamics. The springs of this Thesis are interdisciplinary, bridging state-of-the-art methodologies from numerical analysis, machine learning, microscopic simulations, bifurcation theory and neuroimaging. The research efforts and results were focused in three main directions.

We have first focused on the solution of the source-localization problem in neuroimaging, exploiting and comparing the performance of state-of-the-art regularization methods, namely the standarized Low Resolution Electromagnetic Tomography (sLORETA), the weighted Minimum Norm Estimation (wMNE) and the dynamic Statistical Parametric Mapping (dSPM) and information arising from the electrophysiology of the brain. In particular, the research efforts were focused on the localization of the sources of brain activity of children with epilepsy based on electroencephalograph (EEG) recordings acquired during a visual discrimination working memory (WM) task using numerical regularization algorithms. Importantly, our study and findings reveal also the importance and potential that originates from the use of physics-based information as well as publicly available scientific resources such as the “Neurodevelopmental MRI" database, which allow the researchers to numerically analyse available neuroimaging data and investigate questions beyond the scope of the original studies.

Next, we addressed a computational framework for the embedding of high-dimensional

spatio-temporal data produced by microscopic simulators in low-dimensional manifolds, the identification of the appropriate parsimonious observables based on the constructed low-dimensional models and finally the construction of coarse-grained bifurcation diagrams from spatio-temporal data. Thus, we exploit manifold learning and in particular Diffusion Maps to identify the intrinsic dimension of the manifold where the emergent dynamics evolve and for feature selection. Based on the selected features, we learn the right-hand-side of the effective partial differential equations (PDEs) using Feed-forward Neural Networks (FNNs). Based on the learned black-box model, we construct the corresponding bifurcation diagrams, exploiting the arsenal of numerical bifurcation theory. For our illustrations, we implemented the proposed method to construct the one dimensional bifurcation diagram of the celebrated and well-studied FitzHugh-Nagumo PDEs of activation-inhibition dynamics in neurons from data generated by Lattice Boltzmann simulations.

Finally, we addressed a numerical method based on physics-informed random-projection neural networks for the solution of initial value problems (IVPs) of low-dimensional systems of ODEs with a focus on stiff problems. The numerical solution of the IVPs is obtained by constructing a system of nonlinear algebraic equations, which is solved with respect to the output weights by the Gauss-Newton method. The performance of the proposed scheme was assessed through three benchmark stiff IVPs, namely the Prothero-Robinson, the van der Pol model and the the ROBER problem. Furthermore, the proposed scheme was compared with an adaptive Runge-Kutta method, and a variable-step variable-order multistep solver based on numerical differentiation formulas, as implemented in the `ode45` and `ode15s` of the matlab ODE suite. We show that the proposed scheme yields good approximation accuracy, thus outperforming in some cases `ode45` and `ode15s`, especially in the cases where steep gradients arise. Furthermore, the computational times of our approach are comparable with those of the two matlab solvers for all practical purposes.

Contents

Abbreviations

$4-ExSo-MUSIC$ 4th order Extended Multiple Signal Classification

ANN Artificial Neural Network

BEM Boundary Elements Model

BRE Benign Rolandic Epilepsy

ChNN Chebysev Neural Network

cMEM coherent Maximum Entropy on the Mean

CNT Control-Non Target

CR Conductivity Ratio

CT Control Target

dSPM dynamical Statistical Parametric Mapping

EEG Electroencephalography

ELM Extreme Learning Machine

eLORETA exact Low Resolution Electromagnetic Tomography

ENT Epileptic-Non Target

ERP Event Related Potential

ESL EEG Source Localization

ET Epileptic Target

FDR False Discovery Rate

FEM Finite Elements Model

FHN FitzHugh-Nagumo

fMRI functional Magnetic Reasonance Imaging

FNN Feedforward Neural Network

GE Generalised Epilepsy

IPL Inferior Parietal Lobule

IVP Initial Value Problem

JL Johnson and Lindenstrauss

LeNN Legendre neural Network

MEG Magnetoencephalograph

ML Machine Learning

MNE Minimum Norm Estimation

MRI Mgnetic Reasonance Imaging

NN Neural Network

ODE Ordinary Differential Equation

PDE Partial Differential Equation

PINN Physics Informed Neural Network

QSSA Quasi-Steady State Assumption

RBF Radial Basis Function

RPNN Random Projection Neural Network

RVFLN Random Vector Functional Link Network

SINDy Sparse Identification of Non Linear Dynamics

SLFNN Single Layer Feedforward Neural Network

sLORETA standardized Low Resolution Electromagnetic Tomography

SNR Signal to Noise Ratio

SPL Superior Parietal Lobule

WM Working Memory

wMNE weighted Minimum Norm Estimation

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Recent theoretical and technological advances in the field of numerical analysis methods for the analysis and processing of large-scale data produced either from experiments or from the simulation of microscopic models (e.g. Molecular Dynamics, Monte Carlo, Lattice-Boltzmann simulations) have allowed better modeling and analysis of complex systems for which good models in the form of e.g. ordinary or partial differential (ODEs or PDEs) equations for describing emerging dynamic behaviors are not available. Examples of such complex problems can be found in various areas of science and engineering such as Environmental Engineering, Materials Engineering, Fluid Mechanics, Biology and Biomechanics, Epidemiology and Neuroscience. The importance of understanding the mechanisms that lead to emerging dynamics from the interactions of units (molecules, atoms, neurons) at a microscopic level, led to the awarding of the Nobel Prize in Physics in 2021 to Klaus Hasselman, Giorgio Parisi and Sykyro Manabe for their research work in this specific field.

There has been a massive progress in understanding how the interactions between units in the micro scale give rise to specific emergent behaviors of the system at the macro scale. Examining the complex non-linear dynamics of the system at the macro scale allows for the derivation of fundamental laws about the behavior of the system as a whole. Such a derivation is majorly carried out by using the arsenal of numerical analysis methods, such as finite differences, finite elements, spectral

methods, stiff time integrators, bifurcation analysis and more. However, despite the tremendous progress in the development of numerical algorithms, bridging the physics that may be available in a detailed microscopic scale with the available numerical analysis tools for continuum models for the study of the evolution of the emergent complex dynamics is an open challenging issue. Moreover, real-life physical systems may have missing or noisy boundary conditions that hinder the implementation of traditional approaches. In order to handle these issues, many researchers have turned their attention to the analysis of big-data . The discovery of physical laws from data and consequently the systematic analysis of their dynamics with established numerical analysis techniques is the holy grail in the study of complex systems and has been the focus of intense research efforts over the the last decades [1–6].

Such data can be available either directly from experiments or from detailed simulations using for example molecular dynamics, agent-based models, and Monte-Carlo methods. However, temporal simulations are not sufficient for performing a series of tasks at the macroscopic/emerging level. Such tasks of paramount importance include stability analysis, design of controllers and numerical bifurcation analysis for tracing unstable branches of solutions as well as the identification of critical points in the parameter space that mark the onset of qualitatively different behaviours, including e.g. catastrophic shifts. Thus, all in all, we confront with two major problems: (a) the identification of the appropriate variables that define the emerging (coarse-grained) dynamics, (b) the construction of low-dimensional models based on these variables.

Even in the case when these major problems do not arise, (i.e., when models describing the coarse-grained dynamics are available), obtaining the solution of ODEs and/or PDEs coming from complex systems often turns out to be extremely demanding [7–10]. Specifically, there exists a huge range of complex problems which require dealing with (large-scale systems of) stiff ODEs [11, 12] for their efficient modelling. As a result, the development of new methods that aim to provide efficiently accurate numerical solutions is of paramount importance [13–18]. Machine learning seems to be a promising alternative to the classical numerical algorithms [19–22].

Despite the huge amounts of datasets that are available today (experimental or simulated), most of the times it is not possible to incorporate them seamlessly into the existing physical models. There have been many promising efforts to assimilate these data streams, yet the lack of universally acceptable models as well as the heterogeneity of most of the available data lead to the need for a combinatorial approach. The combination of “black-box" information acquired from data and the a priori physics-informed knowledge can be incorporated by machine learning (ML)[19, 20, 23]. Synergistic integration between ML algorithms and physical models has been advocated for modeling the Earth system [24], biomedical and biophysical processes [25] and more. Hence, the effort of interpreting physical laws and "teaching" the ML algorithms to respect them by providing "informative priors" (theoretical constraints, inductive biases) is needed for enabling our empirical understanding of the world to improve the effectiveness of a learning algorithm.

An enlightening example of this new philosophy is the family of "physics-informed Neural Networks" (PINNs) [20] which are trained to solve supervised learning tasks while respecting the given laws of physics described by nonlinear differential equations. The main motivation for developing these algorithms is that such kind a priori informative constraints can lead to robust results even in the case of imperfect datasets (noise, extreme or missing values etc). Hence, PINNs can generalize accurately with increased performance, due to this extra physical information about the fundamental nature of the under study phenomenon.

Following the same perspective, a-priori physics- informed constraints can be used to improve the performance of the traditional numerical algorithms for challenging tasks such as optimization. Numerical optimization techniques have been developed to deal with many inverse ill-posed problems, which may have an infinite set of possible solutions and the extraction of the desirable one, the one that is justified by the real phenomenon, can be obtained using physical constraints. Usually, these constraints are incorporated by adding extra regularization terms to the cost function [26, 27].

The idea of interpreting a-priori information in order to increase the accuracy and validity of numerical analysis has a particular application in neuroscience and specifically for solving the inverse problem in Electroncephalography Source Localization (ESL). In this problem, which has attracted the interest of many researchers, by exploiting information about the functionality of the brain which has been justi-

fied through experiments, one can identify the actual sources (neurons) that create the available EEG recordings [28, 29]. The physical information about the brain activity is incorporated through various regularization constraints to a traditional optimization algorithm. These synergistic numerical regularization methods have been proved promising techniques for acquiring information about the source space (brain) when only information about the sensor space (scalp) is available. ESL has been applied for identifying brain regions that may be involved in brain disorders and especially epilepsy. One of the most challenging open problems in the field is that of early diagnosis of epileptic crises. Epilepsy affects more than 65 million people worldwide while, approximately 1 out of 150 children is diagnosed with epilepsy during the first 10 years of their life [30]. Although many children self-heal before adulthood, it has been shown that children with epilepsy confront various cognitive and behavioural problems such as problems in learning, attention and memory capacity [31]. Thus, the systematic study of the brain (dys)functionalities of children with epilepsy, and ultimately the development of efficient/targeted treatments is one of the most challenging problems in neuroscience and beyond. In this Thesis, motivated by the aforementioned gaps in literature, we demonstrate in the important case of epilepsy how physics-informed a-priori constraints can be utilized for identifying crucial brain regions from non-invasive brain recordings (EEG) for children. Studies for children are relatively limited but obviously of utmost importance.

Regarding the problem of bridging scales and discovering the hidden physical laws, thus extracting “useful" information from high-dimensional spatio-temporal data for coarse-grained modelling and numerical bifurcation analysis purposes, we propose a three-steps computational framework to (a) perform feature selection based on manifold learning, (b) using machine learning to learn the right-hand-sides of the effective PDEs, and (c) based on the constructed "black-box" model to perform numerical bifurcation analysis.

Finally, motivated by the fact that stiff systems of ODEs frequently arise in the modelling at the coarse-continuum scale of the dynamics complex systems, we provide an alternative of solving low-dimensional systems of ODEs using physics-informed ML tools.

1.2 State of the art

1.2.1 Solving the Inverse Problem in Neuroimaging

The application of the discussed idea in real life systems is of particular importance. One great example, which has caught the attention of the scientific community, is the brain neural system. The field of neuroimaging has began to apply both theoretical and experimental procedures originating in complexity science—usually in parallel with traditional methodologies [32–38] by data produced by non-invasive neuroimaging techniques such as EEG recordings that are widely used for clinical assessment [39–46]. An important weakness though is that sensor-level analysis of brain signals does not give information about the “true" regions of the brain that are being involved. This is the reason why the topic of EEG Source Localization has been widely developed during the past decades (see review [47]). Moving from the surface of the head deep in the brain provides much more information about the brain activity. Of course, from the mathematical point of view, the problem of ESL is an inverse ill-defined problem and cannot be solved exactly. There are many methods that have been developed to deal with this problem [29, 48]. Among the many inverse methods proposed so far, some make use of physiologically relevant a-priori knowledge about both the location and orientation of dipole sources at the origin of signals collected at the scalp. In order to have an accurate 3D tomography of the solution, we have to combine this information with a space model. In this case, the space is the human head and the model is called volume conductor model (the realistic head model obtained by Magnetic Resonance Imaging (MRI) segmentation). This processing considerably increases both the precision of localized sources and the estimation of associated time series, which are analogous to the local field potentials [49, 50]. However, the source localization problem is an ill-defined problem and as such, it poses open questions regarding its robustness and in general the validity of the obtained results [51]. Thus, comparative studies between the various numerical methods that aspire to solve the source localization problem are critical[52, 53].

Toward this aim, Jatoi et al.[54] have compared the sLORETA with the exact LORETA (eLORETA) based on EEG recordings of a visual experiment on healthy subjects. Cincotti et al. [55] compared two techniques for source localization, namely the surface Laplacian and LORETA using EEG recordings from a group of Alzheimer disease patients and age-matched controls. Yao and Devald [56]

compared the performances of several source localization methods on the basis of both simulated and experimental EEG data of somatosensory evoked potentials. Attal and Schwartz [57] compared the performance of three methods, namely the wMNE, sLORETA and the dSPM for the characterization of distortions in cortical and subcortical regions using a realistic anatomical and electrophysiological model of deep brain activity. Seeland et al. [58] compared wMNE, sLORETA and dSPM using EEG data taken from eight subjects performing voluntary arm movements.

Regarding epilepsy, the majority of the studies have performed source localization with the aid of EEG-functional MRI (fMRI) recordings and/or simulated data approximating epileptic spatio-temporal patterns such as spikes and discharges. Fewer studies have dealt with source-level analysis and compared different source localization methods using EEG clinical data taken by children with epilepsy. Among these studies, Adebimpe et al. [59] performed source localization using eLORETA to investigate changes in functional connectivity in children with benign rolandic epilepsy with centrotemporal spikes using resting-state EEG recordings. Groening et al. [60] combined EEG–fMRI and EEG source analysis to identify epileptogenic foci in children. Elshoff et al. [61] examined the efficiency of EEG-fMRI and EEG source analysis to localize the point of seizure onset in children with refractory focal epilepsy.

Ioannides et al. [62] assessed the performance of two source localization methods, wMNE and eLORETA using magnetoencephalography (MEG) signals of ictal and interictal epileptiform discharges in epilepsy and K-complexes. Chowdhury et al.[63] compared the performance of the coherent Maximum Entropy on the Mean (cMEM) and the 4th order Extended Source Multiple Signal Classification (4-ExSo-MUSIC) using MEG and EEG synthetic signals mimicking normal background and epileptic discharges. Hasan et al. [64] evaluated four algorithms (dSPM, wMNE, sLORETA and cMEM) using simulated data from a combined biophysical/physiological model used to generate interictal epileptic spikes as well as real EEG data recorded from one epileptic patient who underwent a full presurgical evaluation for drug-resistant focal epilepsy. Moeller et al. [53] provides a review of the studies that used EEG-fMRI recordings to assess different types of epileptic form activity, underpinning the necessity for comparing with other methods including EEG source analysis.

The above studies have focused mainly on the study of the brain regions that are activated during seizure periods or before their onset. Several other studies have also aimed at analyzing the emerged patterns during seizure periods. For example, Fergus et al. [65] used a supervised machine learning approach to classify seizure and non-seizure records using an open dataset of seizured EEG signals from both children and adults. On the other hand, it has been shown, that studying epileptic seizure-free EEG recordings is of great importance as such analysis can facilitate the identification of patients at risk of epilepsy and/or forecast forth-coming seizures (for a discussion and review of ictal and interictal activity and their analysis see for example [46]).

1.2.2 Bridging Microscopic and Macroscopic Scales: Learning PDES from data

From the early '90s, researchers employed machine learning algorithms for system identification using macroscopic observations, i.e. assuming that we already know the correct variables to model the underlying dynamics and the derivation of normal forms [66–71]. Bongard and Lipson [72] proposed a method for generating symbolic equations for nonlinear dynamical systems that can be described by ODEs from time series data. This approach allows us to model each variable independently and is thus able to extract the less obvious characteristics and simplify the equations during modelling. The method was validated with four simulated and two real systems arising from different fields such as mechanics, ecology and systems biology. The reported results suggest that similar methods can be exploited to automatically identify, testable hypotheses of such complex systems: not just provide information about possible correlations between variables but also automatically provide information about possible causal relationships between them. Brunton et al. [73] addressed the so-called sparse identification of nonlinear dynamics (SINDy) method to obtain *explicit* data-driven PDEs from data when the "good" variables are known, and construct normal forms for bifurcation analysis. Making the rational assumption that there exist only a few important terms that govern the under study dynamics and hence, the equations are sparse in the space of possible functions, they used sparse regression to extract the fewest terms required to effectively describe the observational data. The authors evaluated the method in a wide range of problems: simple canonical systems, the chaotic Lorentz system, the fluid vortex shedding

behind an obstacle and more. The proposed methodology provides an effective tool for handling big datasets where the problem of the curse of dimensionality is inevitable. Extracting the few macroscopic variables is undoubtedly a convenient alternative.

However, for many complex systems, such “good” macroscopic observables that can be used effectively for modelling the dynamics of the emergent patterns are not always directly available. Thus, such “hidden” variables have to be identified from data. In the early 2000’s, the Equation-Free multiscale framework [3, 4, 74, 75] provided a systematic framework for the numerical analysis (stability, design of controllers, optimization, numerical bifurcation analysis) of the emergent dynamics as well as for the acceleration of microscopic simulations, by bridging the microscale where the physical laws may be known and the macroscopic scale where the emergent dynamics evolve. This bridging is achieved via the concept of the “coarse time steppers", i.e. the construction of a black-box map on the macroscopic scale. By doing so, one can perform numerical analysis, even for macroscopically large-scale systems tasks by exploiting the arsenal of matrix-free methods in the Krylov subspace [3, 76–78, 75], thus bypassing the need to construct explicitly models in the form of PDEs.

In the case when the macroscopic variables are not known a-priori, one can resort to state-of-the art manifold learning algorithms such as Diffusion Maps [79, 6, 80, 81], a manifold learning algorithm that can be used either for dimensionality reduction or for feature extraction. Diffusion maps exploit the relationship between heat diffusion and random walk Markov chain. The basic observation is that if we take a random walk on the data, walking to a nearby data-point is more likely than walking to another that is far away. The aim of the algorithm is to identify a low-dimensional embedding of a dataset into the Euclidean space, the coordinates of which can be computed from the eigenvectors and eigenvalues of a diffusion operator on the data. The Euclidean distance between points in the embedded space is equal to the "diffusion distance" between the probability distributions centered at those points [82]. Diffusion maps belong to the family of nonlinear dimensionality reduction methods which focus on discovering the underlying manifold that the data has been sampled from. By integrating local similarities at different scales, diffusion maps give a global description of the data-set. Compared with other manifold learning methods, the diffusion map algorithm is robust to noise perturbation and computationally inexpensive.

Over the last few years, efforts have been focused on developing physics-informed machine learning methods for solving forward and inverse problems, i.e. obtaining solution of high-dimensional multiscale problems described by PDEs and discovering hidden physics [1, 83, 19, 84], thus both identifying the correct observables and based on them to learn the effective PDEs. Lee et al. [5] addressed a methodology to find the right hand sides of macroscopic PDEs directly from microscopic data using Diffusion maps and Automatic Relevance Determination for selecting a good set of macroscopic variables, and Gaussian processes and artificial neural networks for modelling purposes. The approach was applied to identify a "black-box" PDE from data produced by Lattice-Boltzman simulations of the Fitzhugh-Nagumo model in the parametric region where sustained oscillations arise.

1.2.3 Using Machine Learning for the numerical solution of differential equations

On the problem of the numerical solution of differential equations, ML approaches have been proposed as promising alternatives to the traditional methods. Specifically, the idea of using Artificial Neural Networks (ANNs) for the numerical solution of differential equations dates back to the '90s [7–10]. The main idea behind this approach is the use trial functions with the aid of a single-hidden layer ANN trained to minimize the error between the predictions and the right-hand side of the differential equations (a review and presentation of various ANN schemes for the solution of ODEs can be found in Yadav et al. [85]). Lagaris et al. [8] presented a method for solving initial and boundary value problems utilizing ANNs. The proposed method relies on the approximation capabilities of FNNs and leads to a solution that can be written in differentiable, closed analytical form. Training phase includes the minimization of a cost function constituting by two terms, one that satisfies the initial/boundary conditions and contains no adjustable parameters and one which involves a feedforward neural network to be trained so as to satisfy the differential equation. One important feature of the proposed approach is that it enables the computation of the solution at every point of the domain; a feature that is fundamentally different from the traditional numerical schemes.

More recently, scientific advances have renewed the interest for developing and applying physics-informed machine learning algorithms for learning and solving

differential equations [86–88, 14, 89, 1, 90, 20, 91–93]. More precisely the developed data driven algorithms perform supervised learning tasks that comply with the fundamental physical laws underlying the nonlinear systems dynamics. More specifically, due to the fact that stiff ODEs [11, 12] arise in the modelling of an extremely wide range of problems, from biology and neuroscience to engineering processes, material science and chemical kinetics, and from financial systems to social science and epidemiology, there is a re-emerging interest in developing new methods for their efficient numerical solution [13–18].

De et al [13] developed a PINN pipeline to solve accurately a thermal creep flow in a plane channel problem, by utilizing the Theory of Functional Connections (TFC) and constructing the solution of the problem was via Constrained Expressions (CE). The authors solved the Boltzmann equation in the Bhatnagar–Gross–Krook approximation modeling the Thermal Creep Flow in a plane channel. They employed three different kind of ML tools, namely the standard shallow Neural Network (NN), Chebyshev NN (ChNN), and Legendre NN (LeNN) showing that accurate solutions can be obtained with all these networks with a very fast training time. Ji et al solved two classical stiff ODE systems from chemical kinetics by exploiting the Quasi-steady-state assumption (QSSA) [17]. By imposing the QSSA, the number of state variables and transport equations is reduced by eliminating the fast species such that the computational cost can be greatly reduced. Thus, the incorporation of QSSA to a PINN can effectively reduce stiffness. In particular, they incorporate QSSA to a PINN to reduce the stiffness. They applied the PINN algorithm to approximate the solution of the ROBER problem and the POLLU problem. The results revealed that by imposing QSSA on certain species in the kinetic systems and reducing the stiffness, the PINN well captured the dynamic responses of the systems. However, studies have reported severe issues or even failure when dealing with highly stiff problems and/or with problems, the solutions of which exhibit steep gradients. These difficulties stem from various inherent problems linked to the training problem, including for example the estimation of the inner weights and biases of the neural networks and the unbalanced back-propagated gradients (see e.g. the discussion in Wang et al. [92, 94]). Recently, the so-called Extreme Learning Machines (ELMs), have been used to approximate the solution of linear elliptic partial differential equations with sharp gradients with collocation [21] and for the numerical solution of nonlinear PDEs with steep-gradients and furthermore for constructing the corresponding bifurcation diagrams [22]. The keystone idea is

to use a fixed-weight configuration between the input and the hidden layer, fixed biases for the nodes of the hidden layer, and a linear output layer. Hence, the output is projected linearly onto the functional subspace spanned by the nonlinear basis functions of the hidden layer, and the only unknowns that have to be determined are the weights between the hidden and the output layer. Their estimation is performed in one step by solving a (nonlinear) least squares problem.

1.3 Contributions of the Thesis

In this Thesis, the main focus is on the development and implementation of machine learning and numerical analysis methods for the modeling analysis of the emergent low-dimensional dynamics of complex non-linear systems. Based on the above, the main contributions of this Thesis are the following.

The first and major contribution of this Thesis relates to performing coarse grained numerical analysis tasks without having a-priori either the equations of the coarse grained model or the coarse grained variables. More specifically, building on previous work [5], we develop a computational framework based on manifold and machine learning to perform numerical bifurcation analysis from spatio-temporal data produced by microscopic simulators [95]. In particular, in order to discover the appropriate set of coarse-scale variables, i.e. for feature selection we use Diffusion maps, while for the identification of the right-hand side of the apparent coarse-grained PDE over a grid of values of the bifurcation parameter and initial conditions, we use FNNs. For our illustrations we use a Lattice Boltzmanm simulator of the FitzHugh-Nagumo (FHN) spatio-temporal dynamics of activation-inhibition dynamics of neurons. Upon training, the tracing of the coarse-grained bifurcation diagram using the machine-learning regressors is obtained by coupling the pseuso-arc-length continuation condition. The performance of the proposed scheme is compared to the reference bifurcation diagram obtained by finite differences of the deterministic FHN PDEs. The novelty and the contribution of the employed approach dwells upon developing a completely data driven methodology for handling datasets acquired from mechanisms that knowledge about a closed form model does not exist. In many real life systems, one is able to collect (simulated or experimental) data from microscopic observations easily but the macroscale dynamics is very hard to be constructed analytically. Hence, utilizing the proposed approach, we can

systematically use the arsenal of numerical bifurcation analysis theory to study the emergent dynamics in the parameter space.

Regarding the task of the numerical solution of low-dimensional systems of ODEs that may describe the evolution of the emergent dynamics of complex systems, we propose a numerical scheme based on physics-informed Random Projection Neural Networks (RPNNs), for the solution of low-dimensional initial value problems of ODEs. RPNNs are a family of neural networks including randomized and Random Vector Functional Link Networks (RVFLNs) [96–99], Echo-State Neural Networks and Reservoir Computing [100–102], and Extreme Learning Machines [103–105]. Our scheme is implemented by a simple single-layer RPNN with Radial Basis Functions (RBFs) with parameters that are properly uniformly-distributed random variables. The feasibility of the scheme is guaranteed by the celebrated Johnson and Lindenstrauss Lemma [106] and the universal approximation theorems proved for RPNNs and in particular for ELMs [107], thus constituting a Lipschitz embedding constructed through the random projection. To demonstrate the efficiency of the proposed method, we have chosen three benchmark stiff low-dimensional systems of ODEs, namely the Gear-Shampine/Prothero-Robinson ODE [11, 108], for which an analytical solution exists, the well-known van der Pol equations [109] and the ROBER problem, a stiff system of three nonlinear ODEs describing the kinetics of an autocatalytic reaction [110]. The performance of the proposed scheme is assessed in terms of both approximation accuracy and computational times in comparison with two widely used built in functions from the matlab suite, the `ode15s` and the `ode45`. We show that the proposed scheme outperforms `ode15s` in terms of numerical approximation accuracy, especially in the cases where steep gradients arise, while `ode45` in some cases completely fails or needs many points to satisfy a specific tolerance, while results in comparable computational times.

Finally, we assess the performance of well-established regularization approaches exploiting to solve the inverse source localization problem from EEG signals recorded from children with epilepsy during a visual working memory (WM) task [111]. In particular, for the solution of the inverse problem, we use three methods, namely, sLORETA [112], dSPM [113] and wMNE [114]. A statistical comparative analysis between methods and groups (healthy children vs children with epilepsy) revealed the crucial role of the Superior Parietal Lobule (SPL) and Inferior Parietal Lobule (IPL) at WM. Our findings are in line with fMRI studies [115–118] that have

shown that SPL and IPL are being involved in WM processing and thus can serve as a “biomarker" for identifying, monitoring and accessing epilepsy in children. This is the first study to perform a comparative analysis of three numerical methods, namely the sLORETA, wMNE and dSPM to identify differences at the source level between healthy children and children with well-controlled epilepsy (i.e. in the absence of seizures) during a working memory task. In the absence of anatomical MRI scans, we used the publicly available “Neurodevelopmental MRI database" that provides age-specific average MRI templates.

1.4 Organization of the Thesis

The structure of the Thesis is as follows. We begin with the description of the Electroncephalography Source Localization problem and the regularization methods which are used for identifying the source signals in chapter 2. In section 2.1, we formulate the problem and discuss the approach of the regularization techniques. We start by setting the mathematical formulas that describe the functionality of the brain that leads to the recorded EEG signals. Next, we construct the mathematical model that is used for obtaining the solution of the inverse problem which reveals the true brain regions that are responsible for the measured EEG data. As we have already described, this is achieved by numerical optimization techniques with regularization (section 2.2). In sections 2.2.1, 2.2.2 and 2.2.3, we describe the three inverse EEG methods that we are going to apply for the source identification, namely sLORETA, wMNE and dSPM, their assumptions and their inverse solutions. In chapter 3, we present the results of the source localization analysis in epileptic children during a visual WM task. We describe the appropriate preliminaries in section 3.1 and in section 3.2 we provide all the necessary information about the experimental procedure. The subjects, the experiment itself (design and stimuli), the EEG recordings and the pre-processing are written in detail during the sections 3.2.1-3.2.4. Furthermore, the appropriate age-specific head models that were used for solving the forward problem are described in section 3.2.5. The results of our comparison between the two groups (epileptic and control children) are shown in section 3.3.1. We describe the employed statistical tests, extracting the differences between the two groups and revealed usefull information about the IPL and its role in the WM. We conclude by discussing the results in section 3.4.

In chapter 4, we describe the machine and manifold learning algorithms used in this Thesis. In section 4.1, we present the FNNs and discuss briefly their approximation capabilities (section 4.1.1). Next, in section 4.2 we discuss the the basic concept behind the use of RPNNs; we also discuss briefly the celebrated Johnson and Lindenstrauss Lemma [106] and in section 4.3, we describe the manifold learning algorithm of Diffusion Maps.

In chapter 5, we describe the proposed machine learning algorithm for the solution of low-dimensional ODEs. We state the problem and provide some preliminaries on the solution of IVPs with FFNs. In Section 5.2, we describe our approach for the solution of IVPs of ODEs with the use of RPNNs, providing also a pseudo-code of the corresponding algorithm (section 5.2.2), and discuss its approximation properties within the framework of the universal approximation theorem of ELMs [107] (section 5.2.1). Section 5.3 presents the performance of the proposed method for solving (stif) low-dimensional ODEs with PINNs. After that, we present the results regarding applications, namely the Prothero Robinson problem (5.3.1), the van der Pol problem (5.3.2) and the ROBER problem (5.3.3). Results and their conclusions are discussed in section 5.4.

In chapter 6, we present the numerical machine learning-based method for constructing coarse-scale bifurcation diagrams from data coming from microscopic simulators. In section 6.1 we state the problem of identifying the coarse grained dynamics from data by identifying the effective macro scale variables. The feature selection method for selecting the right variables is described in section 6.1.1. The macroscale model of the FHN two coupled reaction diffusion PDEs is illustrated in section 6.1.2. The detailed spatio-temporal data were produced by Lattice Boltzmann simulations. We discus briefly the processing of obtaining the coarse-scale variables and their spatial derivatives, which we use for modeling the coarse-grained dynamics. Finally, we summarize and provide two short abstract pseudo-code algorithms for the proposed pipeline in section 6.1.4. In section 6.2, we perform a systematic numerical bifurcation analysis for the closed form FHN PDEs and construct the bifurcation diagram which we consider as the reference bifurcation diagram for the later comparisons. Results regarding the convergence of the reference bifurcation diagram are also shown. In section 6.3, we illustrate the results of the proposed framework for constructing bifurcation diagrams directly from observations for two different approaches: (a) without feature selection (6.3.1) and (b) with feature selection (6.3.2). We demonstrate in detail the performance of the training phase

and describe the application of the arc-length numerical continuation method in the "black-box" machine learning models. Finally, we discuss the results in section 6.4.

Finally, in chapter 7 we report the conclusions of the above described methodologies. We discuss the limitations and the gaps that have emerged until now as well as some future directions that have to be investigated in the process.

Chapter 2

Electroencephalography Source Localization (ESL)

2.1 Mathematical formulation

Source localization aims at identifying the (unknown) sources of the brain from data taken usually from noninvasive electromagnetic recordings (here: EEG recordings). Its solution involves a forward and an inverse problem. The forward problem refers to the calculation of the electric potentials of the electrodes starting from a given electrical source. The solution of the forward problem is related to the construction of a head model. The head model contains both anatomical information and the conductivities of three layers, namely the skull, the cortex and the scalp. Anatomical images can be obtained experimentally with the aid of MRI scans, while volume conduction models can be constructed using e.g. the Boundary Elements Model method (BEM) [119] or the Finite Elements Model method (FEM)[120]. The head model volume is tessellated into small-sized cubes, the voxels. Sources may be associated to single voxels or clusters of voxels. Here, each voxel is asscociated to a single source. The relation between the scalp recordings and the discretized head model volume is performed using the linear matrix equation:

$$V = Gx + \varepsilon, \tag{2.1}$$

where V is a known $N \times 1$ matrix which contains the time instances as recorded by each channel (N is the number of channels), x is the unknown $M \times 1$ matrix of the intensities of the M sources (M is the number of voxels).

The matrix G, with dimensions $N \times M$ is the so-called lead field matrix that contains the information of the head geometry and conductivities. G is known (from the solution of the so called forward problem (see e.g. in [121])) and is related with the head model [122]; ε reflects the noise in the measurements.

The inverse problem is ill-defined, as there is an infinite number of combinations of positions and intensities that could effectively produce the electric potentials and magnetic fields measured. Regularization is a common scalarization method used to solve similar problems as in 2.1. One form of regularization is to minimize the weighted sum of the objectives:

$$\hat{x} = \min_{x}(||V - Gx|| + a||x||) \tag{2.2}$$

where $a > 0$ is a problem parameter. As a varies over $(0, \infty)$, the solution traces out the optimal trade-off curve.

Another common method of regularization, especially when the Euclidean norm is used, is to minimize the weighted sum of squared norms, i.e.,

$$\hat{x} = \min_{x}(||V - Gx||_2^2 + a||x||_2^2), \tag{2.3}$$

for various values of the parameter $a > 0$. These regularized approximation problems each can solve such kind of problems as in 2.3 making both $||V - Gx||$ and $||x||$ small, by adding an extra term or penalty associated with the norm of x.

Regularization is used in several contexts. In an estimation setting, the extra term penalizing large $||x||$ can be interpreted as our prior knowledge that $||x||$ is not too large. In an optimal design setting, the extra term adds the cost of using large values of the design variables to the cost of missing the target specifications. The constraint that $||x||$ must be small can also reflect a modeling issue. It might be, for example, that $V = Gx$ is only a good approximation of the true relationship $V = f(x)$ between x and V. In order to have $f(x) \approx V$, we want $Gx \approx V$, and also need x small in order to ensure that $f(x) \approx Gx$.

We will see in the below sections that regularization can be used to take into account variation in the matrix G. Roughly speaking, a large x is one for which variation in G causes large variation in Gx, and hence should be avoided.

The most common form of regularization is the based on the above equation, with Euclidean norms, which results in a (convex) quadratic optimization problem (Tikhonov regularization problem [123, 124]):

$$\hat{x} = \min_x(||V - Gx||_2^2 + a||x||_2^2) = x^T(G^TG + \alpha I)x - 2V^TGx + V^TV. \tag{2.4}$$

This Tikhonov regularization problem has the analytical solution:

$$\hat{x} = (G^TG + \alpha I)^{-1}G^TV. \tag{2.5}$$

Since $G^TG + \alpha I > 0$ for any $\alpha > 0$, the Tikhonov regularized least-squares solution requires no rank (or dimension) assumptions on the matrix G.

The idea of regularization, i.e., adding to the objective a term that penalizes large x, can be extended in several ways. One useful extension is to add a regularization term of the form $||Wx||$, in place of $||x||$. In many applications, the matrix W represents an approximate differentiation or second-order differentiation operator, so ||Wx|| represents a measure of the variation or smoothness of x [48]. We can also add several regularization terms or different norm for the regularization term.

So, in its general form a linear optimization problem with regularization can be read as:

$$\hat{x} = \min_x(||V - Gx||_2^2 + \sum_{i=1}^{k} a_i||W_ix||_p). \tag{2.6}$$

In the above, k is the number of regularization constraints (refelcting the a-priori physiological information); the matrix W, $M \times M$, is a weighted matrix related to the imposed constraints; α_i is the regularization parameter and denote the importance of every constraint. For different choices of W, k and p (reflecting the type of the norm), we get different methods.

2.2 Tikhonov regularization techniques for solving the ESL.

2.2.1 Weighted Minimum Norm Estimation (wMNE)

For $W = I$ (the identity matrix) and $p = 2$ (the $L-2$ norm) in 2.6 we get the Minimum Norm Estimation (MNE) [125], the one written in 2.5. MNE uses the mathematical assumption that the best solution, through the infinite set of solutions, is the one with the minimum norm. Despite the fact that MNE was the first method used to extract a 3D distributed solution, the simplicity of its assumption often leads to inadequate solutions. In particular, it has been shown, that this method fails in identifying deep sources [126]. Because of the minimum norm constraint, sources that are located in deep regions are moved closer to the cortex.

The wMNE method is a variation of the MNE that improves the problem of the mislocation of the deep sources. wMNE uses instead of the identity matrix, a diagonal matrix W_c that contains the weighting factors. From the multiple choices that can be chosen as weighted factors, usually $W_c = diag(||G_i||_2)$ (for i =1,...,M) is chosen [47]. Then, the unique solution is given by:

$$x_{wMNE} = LV, \tag{2.7}$$

where $L = G^T(G^TG + \alpha W_c)^{-1}$ is called the inverse operator with dimensions ($M \times N$).

2.2.2 The dynamic Statistical Parametric Mapping (dSPM)

The dSPM [113] is similar to the wMNE but uses a different regularization. dSPM computes the source estimates of the noise based on the noise covariance matrix $C_\varepsilon = \alpha H$ and normalizes the rows of the inverse operator.
$H = I - \frac{1^T 1}{1^T 1}$ is the centering matrix and plays the role of the identity matrix in the measurement space. Then, from equation 2.7, the source estimates of the noise form a diagonal matrix:

$$C_{\hat{x}} = W_{dSPM} = LC_\varepsilon L^T. \tag{2.8}$$

Thus, the dSPM solution is given by:

$$x_{dSPM} = L_{dSPM} V, \tag{2.9}$$

where $L_{dSPM} = W_{dSPM} L$.

2.2.3 Standarized Low Resolution Electromagnetic Tomography (sLORETA)

sLORETA considers another source of variance, except from the covariance of the measurement noise C_ε: the covariance of the actual sources $C_x = I$. Assuming that the activity of the actual sources and the noise of the measurements are uncorrelated and based on the linear relation of equation 2.1, we have:

$$C_V = GC_xG^T + C_\varepsilon = GG^T + \alpha H. \tag{2.10}$$

Substituting equation 2.10 to 2.7, and taking into account the linear relation of equation 2.7, we can estimate the variation of the estimated sources as:

$$C_{\hat{x}} = LC_V L^T = L(GG^T + \alpha H)L^T = G^T(GG^T + \alpha H)^{-1}G. \tag{2.11}$$

The covariance of the estimated sources is equivalent to the Backus and Gilbert resolution matrix [127], which is given by plugging equation 2.1 into 2.7 and substituting the inverse operator to get:

$$\hat{x} = LGx = G^T(GG^T + \alpha H)Gx = Ax = C_{\hat{x}}x, \tag{2.12}$$

where $A = LG$ is the resolution matrix.

In this case, the solution is given by:

$$x_{sLORETA} = L_{sLORETA} V, \tag{2.13}$$

where $L_{sLORETA} = AL$.

Chapter 3

EEG Source Localization analysis in epileptic children during a visual Working-Memory task.

3.1 Introduction

We perform a source-localization analysis of the brain activity of well-controlled epileptic children during a visual WM task [111]. WM is commonly viewed as a functional integration system with limited capacity that is able to store information within a short-term register and simultaneously manipulate it on-line. Thus, WM is one of the most important components of information processing and its dysfunction leads to various problems in several cognitive functions including mental arithmetic [128], reading [129, 130], decision making [131] and reasoning [132]. Epilepsy affects a lot the WM functioning as it has been shown by many studies [133–137]. Here, for the solution of the inverse problem, we use three methods, namely, sLORETA [112], dSPM [113] and wMNE [114]. A statistical comparative analysis between methods and groups (healthy children vs children with epilepsy) revealed the crucial role of the SPL and IPL at WM. Our findings are in line with fMRI studies [115–118] that have shown that SPL and IPL are being involved in WM processing and thus can serve as a “biomarker" for identifying, monitoring and accessing epilepsy in children.

3.2 Experimental procedure

3.2.1 Subjects

In the study group, 21 children with established childhood epilepsy (age 6—16 years old; mean 11.43 years, SD $\pm$2.3, 10 boys) were enrolled. These children were diagnosed with one out of two following epilepsy syndromes: benign rolandic epilepsy (BRE) (n = 9) and idiopathic generalized epilepsy (GE) (n = 12, including childhood absence epilepsy (n = 5) and generalized epilepsy with tonic-clonic seizures (n = 7)). All children were admitted to the neurophysiology laboratory of the University Hospital of Leuven for a 24-h video-EEG monitoring during which the Event Related Potential (ERP) study was done. They had no anti-epileptic treatment (n = 3) or were on standard anti-epileptic medication (monotherapy, n = 15, duotherapy, n = 3), with drug dosages always being within normal ranges. Patients on monotherapy received valproic acid (n = 7), carbamazepine (n = 4), lamotrigine (n=3) or sulthiame (n = 1). Patients on duotherapy received different drug combinations [43]. None of the patients had structural brain abnormalities; in 18 patients, brain MRI was performed showing normal findings in all cases. Only patients with at least an eight days seizure-free period preceding the test were included [43]. Thereby we could avoid an acute effect of epileptic seizures on the child's performance. All children followed mainstream school and none had a history of learning problems. As a control group, 25 age matched non-epileptic children (mean age 10.76 years, SD $\pm$3.4, 17 boys) were selected, who did not have any school problem either. The study protocol was approved by the Ethical Committee of University Hospital of Leuven. The experimental procedure was performed by Myatchin et al. and for more details please see [43].

3.2.2 Design and Stimuli

The event-related potentials study was done as part of video-EEG monitoring. A visual one-backmatching working memory task was performed: children observed a continuous stream of seven different figures presented one after the other in pseudorandom order at the middle of a computer monitor, which was located at a distance of 1.0m from the subject's eyes. Everyday figures were used (horse, wardrobe, jacket, cake, comb, bunch of grapes, hammer), white with a black contour

on grey background, size 7.5 cm $\times$ 6.5 cm, visual angle $4°18' \times 3°02'$. Each stimulus was presented for 1.5 s, followed by a delay of 1.0 s, after which the next stimulus was presented. During the delay period a fixation point (dark-grey cross) was shown at the middle of the screen to facilitate eyes fixation. Any figure identical to the one immediately preceding it was defined as a target stimulus (probability 0.30). Children were asked to respond to all targets by pressing a button with their dominant hand. Both accuracy and speed were stressed. The single experimental block contained 120 trials, 36 of which were targets. The duration of the block was 5 min. This is an easy working memory task, which was chosen to ensure a good level of participant's performance.

First, the electrode placement and impedance calibration was performed. After that, the experimental procedure was described to the child. The child was seated comfortably in a dimly lit registration room and was instructed to look at the middle of the computer screen placed in front of him to avoid unnecessary eye movements; a fixation point (dark-grey cross) was shown between figures to facilitate eye fixation. The child was also instructed to avoid movements to reduce muscle artifacts in the EEG signal. The instruction for the task was given directly before the task. During the experiment, no interaction with the experimenter was allowed during the task and the experimenter sat out of sight of the child.

3.2.3 EEG recordings

Nineteen Ag/AgCl electrodes (Technomed Europe) were placed according to the international 10-20 system at Fp1, Fp2, F3, F4, F7, F8, Fz, C3, C4, Cz, T3, T4, T5, T6, P3, P4, Pz, O1 and O2. Placement of additional four EOG electrodes resulted in two EOG channels: horizontal EOG – two electrodes on the outer canthi of eyes, and vertical EOG – two electrodes above and below one eye. EOG channels allowed us to detect both vertical and horizontal eye movements in order to effectively remove them from EEG recording during subsequent preprocessing of the signal (see below). Two linked mastoid electrodes were used as a reference. EEG was sampled at a frequency of 1000 Hz with 12 bits A/D converter and amplified using a band-pass filter of 0.095 – 70 Hz. Notch filter was off. Registration of the digital EEG was made using the software program BrainlaB 4.0 (OSG, Belgium). The impedance of all electrodes was monitored for each subject prior to recording and was always kept below 5 kΩ.

3.2.4 Data pre-processing

Data pre-processing was performed offline using the EEGLAB v.5.02 toolbox (Matlab 7.0.4 platform) [138]. The ECG channel was factored out. Data were filtered with a 50 Hz digital low pass filter. Eye movement artifacts were marked and removed from the continuous signal without affecting the signal itself using an ICA-based algorithm [43]. EEG fragments containing movement artifacts as well as any epileptic activity were removed based on visual inspection of the data. This resulted in an EEG signal clean from (eye) movement artifacts and epileptic activity, which was then used for further analysis.

Afterwards, the continuous EEG signal was epoched according to the type of stimulus (Target and Non Target), with 200 ms pre-stimulus (delay period) and 400 ms poststimulus (presentation period of the second stimulus, where the motor responses had not yet taken place). Omitted Target trials (i.e. trials without correct motor response) and committed Non Target trials (i.e. trials with a wrong motor response) were excluded from the analysis. We then performed a down-sampling at 500 Hz and we applied a baseline correction by subtracting the mean value of the 200 ms of the pre-stimulus period. Overall, we ended up with 92 datasets (21 epileptic $\times$ 2 trial types + 25 control $\times$ 2 trial types) of 19 multi timeseries, which were divided into four group types (Epileptic- Target (ET), Epileptic-Non Target (ENT), Control-Target (CT), Control-Non Target (CNT)).

3.2.5 Head models for children

In our study, we did not have individual MRI scans for each child that participated to the experiment. Thus, in the absence of such specific information, we used age-specific MRI templates for children acquired from the “Neurodevelopmental MRI database" [139–142]. The goal of this database is to provide for research purposes, exactly in the absence of specific MRI scans, a series of age-appropriate average MRI reference templates and related information. Each template was constructed using identical procedures to facilitate comparisons across lifespan. The database consists of average templates (T1W and T2W), segmenting priors, and stereotaxic atlases [139]. The “Neurodevelopmental MRI Database” is available online (http://jerlab.psych.sc.edu/NeurodevelopmentalMRIDatabase/). The data-base is publicly available to researchers upon request for clinical and experimental studies of normal

and pathological brain development. The data is shared under a Creative Commons Attribution-NonCommercial-Noderivs 3.0 Unported License (CC BY-NC-ND 3.0; http://creativecommons.org/licenses/by-nc-nd/3.0/deed.en$_$US).

Using this database, we were able to construct an “average" age-specific head model for each child taking into account its age. For our study, we constructed 11 averaged head models (taking into account the database with head models of children between 6 and 16 years old, i.e. one “average" head model per year). In table 3.1, we provide information about the total number of MRI scans per age. Here, for the construction of the head models, as skull conductivities are age-dependent,59 we used different conductivities ratios (CR, cortex/skull) for every age-dependent model. The conductivity value for scalp and cortex was set to the standard value of 0.33 S/m.60 Table 2 presents analytically the different conductivity ratios for every age.

Age	6	7	8	9	10	11	12	13	14	15	16
1.5T	27	27	46	46	62	31	37	34	32	32	34
3.0T	10		19		16		15	11	30		13
Combined	37	27	56	46	72	31	47	34	42	32	44

Table 3.1 Total number of scans per age for 1.5T, 3.0T and combined average MRI templates. All 1.5T MRIs and part of 3.0T MRIs are included in the "Combined" column as in the original publications [139, 141]

Here, for the construction of the head models, as skull conductivities are age-dependent [143], we used different conductivities ratios (CR, cortex/skull) for every age-dependent model. The conductivity value for scalp and cortex was set to the standard value of 0.33 S/m [144]. Table 3.2 presents analytically the different conductivity ratios for every age [144].

Age	6	7-8	9-10	11-12	13-14	15-16
CR	15	20	30	40	50	60

Table 3.2 Conductivity ratios (cortex/skull) for every age-dependent head model. The standard conductivity value for scalp and cortex was set to 0.33 S/m [144].

3.3 Results

For our analysis, we used the BrainStorm toolbox for matlab [145]. Source-reconstructed time series were obtained by combining the EEG recordings with

the appropriate (respect to the age of the subject) constructed MRI templates. From each template, we extracted three layers (scalp, inner skull, outer skull) and the source space (cortical surface). The number of vertices for each layer were set to 2562 vertices for each surface. Then, the volume conduction models were constructed in openMEEG software [146] with BEM. The space resolution for the source model was set to 5124 voxels with fixed orientation perpendicular to the cortex surface.
Thus, the time series at the source level were reconstructed using wMNE, dSPM and sLORETA. The noise was computed from the raw EEG data using the pre-stimulus period for baseline correction and then the noise covariance matrix was calculated. A parameter that has to be determined is the “signal to noise ratio" (SNR). In Brainstorm, the computation of SNR is performed as in the original MNE software of Hamalainen [147]. The signal covariance matrix is “whitened" by the noise covariance matrix and the square root of the mean of its spectrum yields the average amplitude of SNR. The default value in Brainstorm is set to 3.
The main results of source localization procedure are presented analytically at table 3.3. For our illustrations, we have split the time period to three main intervals: the pre-stimulus period [-200ms 0ms), the period exactly after the stimulus [0ms - 199ms] and the post-stimulus period [200ms - 400ms). Our analysis reveals similar results when applying the different methods.

3.3.1 Hypothesis testing using T-test

After the data pre-processing and the implementation of the source-localization algorithms, as described in the previous section, we got 92 time-series at the source space. Our data have a spatio-temporal structure: number of voxels (spatial dimension) and time points (time dimension). In order to perform two-sample T-tests for the identification of statistically significant differences in the activity of the sources among groups, we checked three basic assumptions. In particular, we checked if (1) the amplitude of the source signal at each voxel and at each time instant follows a normal distribution among subjects in a group, (2) the variances of the amplitude of the source signal at each voxel and at each time instant of CT, ET and CNT,ENT are equal, and (3) the amplitudes of the source signal at each voxel and at each time instant are independent for CT, ET and CNT, ENT.
For the test of normality, we used the Shapiro-Wilk test [148]. The null hypothesis

Time Period	Method	CT	CNT	ET	ENT
Pro-Stimulus (-200ms - -1ms)	**wMNE**	-Right Occipital Lobe	-Occipital Lobe	-Right Occipital Lobe	-Occipital Lobe
	dSPM	-Right Occipital Lobe	-Right Occipital Lobe	-Right Occipital Lobe	-Occipital Lobe
	sLORETA	-Right Occipital Lobe	-Occipital Lobe -Left Parietal Lobe	-Right Occipital Lobe	-Right Occipital Lobe
Exactly after Stimulus (0ms - 199ms)	**wMNE**	-Occipital Lobe	-Right Occipital Lobe	-Occipital Lobe	-Superior Parietal Lobe -Occipital Lobe
	dSPM	- Right Occipital Lobe	-Right Occipital Lobe	-Occipital Lobe	-Superior Parietal Lobe -Right Occipital Lobe
	sLORETA	-Right Occipital Lobe	-Occipital Lobe	-Occipital Lobe	-Superior Parietal Lobe -Occipital Lobe
Post-Stimulus (200ms - 399ms)	**wMNE**	-Parietal Lobe	-Right Parietal Lobe	-Right Parietal Lobe	-Right Parietal Lobe
	dSPM	-Parietal Lobe	-Parietal Lobe	-Right Parietal Lobe	-Parietal Lobe
	sLORETA	-Parietal Lobe	-Parietal Lobe	-Right Parietal Lobe	-Parietal Lobe

Table 3.3 Group averaged sources as obtained by the three methods: wMNE, dSPM and sLORETA. CT: Control Target, CNT: Control non-Target, ET: Epileptic Target, ENT: Epileptic non-Target.

of the test is that a sample comes from a normal distribution. The test statistic reads:

$$W = \frac{(\sum_{i=1}^{n} \alpha_i x_{(i)})^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2}, \tag{3.1}$$

where $x_{(i)}$ is the i-th order statistic, $\bar{x}$ is the sample mean and coefficients α_i are given by $(\alpha_1, \alpha_2, ..., \alpha_n) = \frac{m^T V^{-1}}{C}$. C is a vector norm $C = ||V^{-1}m|| = (m^T V^{-1} V^{-1} m)^{\frac{1}{2}}$, $m = (m_1, m_2, ..., m_n)^T$ are the expected values of the order statistics of independent and identically distributed random variables sampled from the standard normal distribution and V is the covariance matrix of those normal order statistics. F-tests were performed to validate the second assumption (i.e. the equality of the variances of the amplitude values of each voxel between ET and CT and between ENT and CNT).

Thus, we tested for normality and equality of variances for each voxel and each time sample (i.e. we have performed a total of $5.148 \times 600 = 3.088.800$ t-tests). The level of significance was set to $p < 0.05$, meaning that the risk of taking a false positive is 5% of the cases. Here, in order to deal with the multicomparison problem,

we used the false discovery rate (FDR) correction [149]. Following this procedure, the null hypothesis could not be rejected. Similarly, the F tests validated also the second assumption. The independence is reasonably assumed to hold true.

Thus having guaranteed that the T-test can be applied, we proceeded with the comparisons ET vs CT and ENT vs CNT. The null hypothesis H_0 for both comparisons was that the two groups have equal means regarding the emerged spatio-temporal activation at the source level. So, we performed $3.E^6$ simultaneous two-sample T-tests with $p < 0.05$ level of significance. FDR was used to deal with the multicomparison problem. Another constraint that we added to avoid spurius and random effects was the one of the minimum duration of the activations. Thus, we excluded all the signals that were statistically significantly for time intervals less than 50 ms.
This statistical analysis revealed that all methods gave relatively similar results.

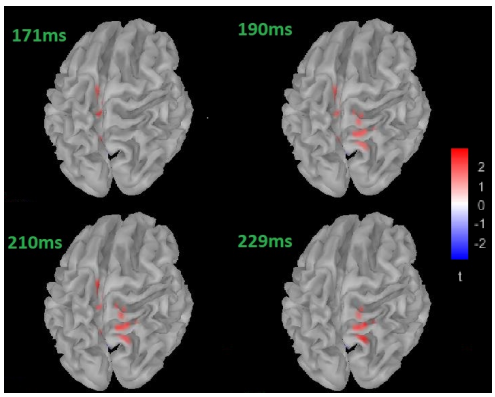
The pair T-test between ENT and CNT revealed a statistically significant difference in the time-range of 170ms-230ms. In this range the the SPL was activated more in the ENT group; the activation of the SPL was mostly at the right hemisphere (figure 3.1).

The pair T-test between ET and CT revealed a statistically significant difference in the time-range of 160ms-360ms. In this range the IPL was activated more in the ET group (only at the right hemisphere) (figure 3.2).

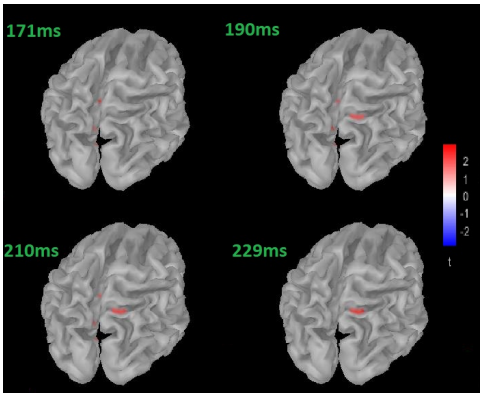
The above finding are summarized in Table 3.4.

Comparison	**Time period**	**Group activated more**	**Region**	**Comments**
ENT vs CNT	170 to 230 ms	ENT	SPL	Only at the right hemisphere.
ET vs CT	160 to 360 ms	ET	IPL	Only at the right hemisphere

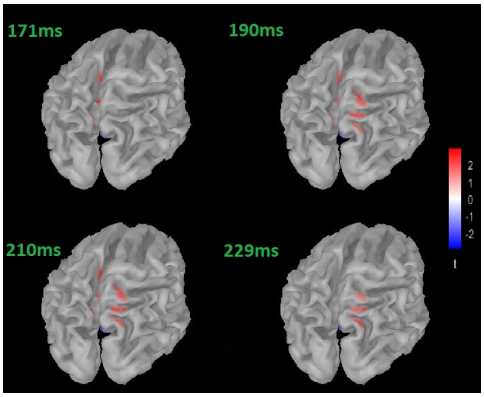
Table 3.4 Analytical presentation of the differences between the comparisons ET vs CT and ENT vs CNT.



(a) wMNE.

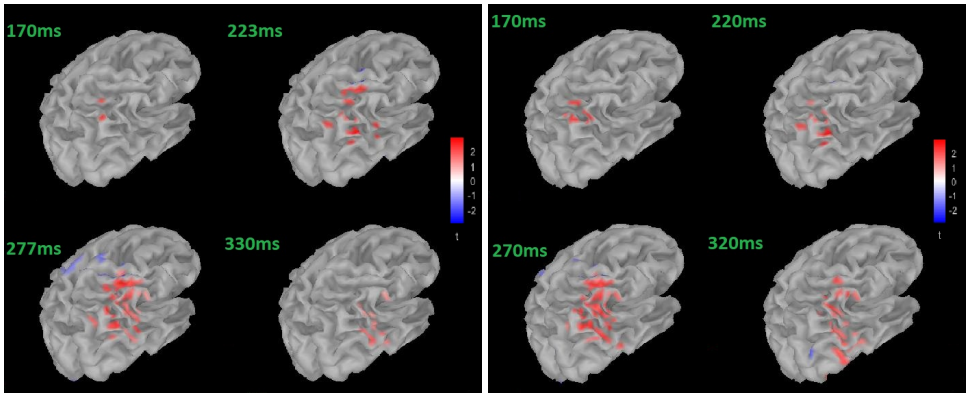


(b) dSPM.

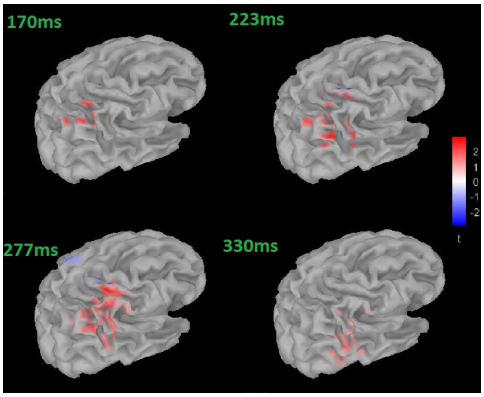


(c) sLORETA.

Fig. 3.1 ENT vs CNT: SPL mainly of the right hemisphere activate more for the ENT group at the time interval from 170 to 230 ms.



(a) wMNE. (b) dSPM.



(c) sLORETA.

Fig. 3.2 ET vs CT: IPL of the right hemisphere activates more for the ET group at the time interval from 160 to 360 ms.

3.4 Discussion

IPL have shown from many studies that can be related to seizures like epilepsy [150], [151] and also to working memory processing [152], [153], [154].

It has been shown, that Parietal lobes play a crucial role in WM. This fact comes from various of studies that have been used fMRI and ERPs. Berryhil et al. [155] studied patients with unilateral right parietal lobe damage in several visual WM recognition tasks and showed that this damage leads to impaired perfomance. Jones et al. [156] observed interactions between WM task difficulty and participants' WM capacity with parietal stimulations. Osaka et al. [157] investigated neural substrates for focusing attention in working memory and found significant activations at left SPL. Li et al. [158] studied the role of parietal lobes in the scope and control of

attention during a visual WM task. Tseng et al. [159] proposed that signal complexity via coherent timing and phase synchronization within the bilateral parietal network is crucial for successful visual WM functioning.

Here, we provide an extra evidence that parietal lobes indeed participate to the WM processing and furthermore epilepsy affects the way with this region is being involved. This study can serve as "biomarker" for identifying, monitoring ans assessing epilepsy in children.

ESL is a very promising technique for studying and identify non - invasively the brain functions. Though, the problem of the sparse space resolution remains and the challenge of reducing its effect to the results is crucial. High density scalp recordings have been noticed to improve significantly the accuracy of the inverse solution. However, this question is still under studying with a lot of promising results [160–162]. In the present study, we have a relatively small number of recordings (19 electrodes), a fact that makes our results sensitive to spacious noise. It would be important, as en extra validation to have similar results with higher density EEG recordings.

Chapter 4

Physics Informed Machine/Manifold Learning methods.

4.1 Feedforward Neural Networks (FNNs)

Neural Networks constitute a biologically-inspired powerful machine learning tool with a huge variety of applications. Their capability of performing complicated tasks made them very rapidly one of the most popular machine learning algorithms for regression, classification, forecasting and many more tasks. They usually contain one input, one output and one or more hidden layers. Every layer has several units (neurons), often fully connected by weights (ω) and biases (b), and an activation function ($\psi(\cdot)$).

Consider a function $f : x \in \mathbb{R}^K \to y \in \mathbb{R}$, where K is the dimension of the input. Hence, one can write explicitly the approximation of a single hidden layer FNN (i.e. the output of the network) with H hidden units as:

$$y = \hat{f}(x; W^h, b^h, \omega^o, b^o) = \sum_{i=1}^{H} \omega_i^o \psi(\omega_i^h \cdot x + b_i^h) + b^o \tag{4.1}$$

where $\omega^o = (\omega_1^o, \omega_2^o, \ldots, ...\omega_H^o) \in \mathbb{R}^{1\times H}$ are the external weights connecting the hidden layer and the output and $b^o \in \mathbb{R}$ is the bias of the output, while the matrix $W^h \in \mathbb{R}^{H\times K}$ with rows $\omega_i^h \in \mathbb{R}^K$ and $b^h = (b_1^h, b_2^h, \ldots, b_H^h) \in \mathbb{R}^H$ are the weights connecting the input and the hidden layer and the biases of the hidden layer,

respectively. In the same way, one can easily extend the above explicit formula for FNNs with more than one hidden layer.

Then, a cost function must be specified, usually the root-mean-square error:

$$E_D = \frac{1}{M}\sum_{j=1}^{M}(u_t^{(j)} - \hat{f}(x^{(j)}))^2, \tag{4.2}$$

where M is the total number of observations, i.e. the pairs $(x^{(j)}, u_t^{(j)})$ such that $u_t^{(j)} = f(x^{(j)})$.

The main task of a neural network is the generalization (i.e. how good it is at learning from the given data and applying the learnt information elsewhere) and the lack of this feature is usually due to overfitting. Foresee and Hagan ([163]) showed that adding the regularization term $E_\omega = \sum_{j=1}^{H}\omega_j^2$ to the cost function will maximize the posterior probability based on Bayes' rule. Hence, the total cost function is:

$$E_{total} = E_D + \beta E_w, \tag{4.3}$$

where β is the regularization parameter that has to be tuned.

4.1.1 FNNs as universal approximators.

Many results are available concerning the approximation properties of FFNs. The most important one from the numerical point of view is the Universal Approximation Theorem, for which we refer to the original papers [164–168].

Theorem 1 (Universal Approximation Theorem)**.** *Let I_n denote the n-dimensional unit cube, $[0,1]^n$. The space of continuous functions on I_n is denoted by $C(I_n)$. Let $\psi(\cdot)$ be any any continuous discriminatory function. Then finite sums of the form:*

$$N(x) = \sum_{i=1}^{H}\omega_i\psi(\omega_i \cdot x + b_i)$$

are dense in $C(I_n)$. In other words, given any $f \in C(I_n)$ and $\varepsilon > 0$, there is a sum, $N(x)$, of the above form for, which

$$|N(x) - f(x)| < \varepsilon \quad \forall x \in I_n.$$

We say that a function $\psi(\cdot)$ is discriminatory if for a measure $\mu \in M(I_n)$ (the space of finite, signed regular Borel measures on I_n): $\int_{I_n} \psi(\omega \cdot x + b) d\mu(x) = 0, \quad \forall \omega \in \mathbb{R}^h$ and $b \in \mathbb{R}$ implies that $\mu = 0$.

Proof. Let $S \subset C(I_n)$ be the set of functions of the form $N(x)$ as above. Clearly S is a linear subspace of $C(I_n)$. We claim that the closure of S is all of $C(I_n)$.

Assume that the closure of S is not all of $C(I_n)$. Then the closure of S, say R, is a closed proper subspace of $C(I_n)$. By the Hahn-Banach theorem, there is a bounded linear functional on $C(I_n)$, call it L, with the property that $L \neq 0$ but $L(R) = L(S) = 0$.

By the Riesz Representation Theorem, this bounded linear functional L is of the form

$$L(h) = \int_{I_n} h(x) d\mu(x)$$

for some $\mu \in M(I_n)$, for all $h \in C(I_n)$. In particular, since $\psi(\omega_i \cdot x + b_i)$ is in R for all ω and b, we must have that:

$$\int_{I_n} \psi(\omega_i \cdot x + b_i) d\mu(x) = 0$$

for all ω and b. However, we assumed that $\psi(\cdot)$ was discriminatory so that this condition implies that $\mu = 0$ contradicting our assumption. Hence, the subspace S must be dense in $C(I_n)$. □

Korning et al [165, 166] proved using the Stone-Weierstrass Theorem also for the case of multilayer FNNs the capability of universal approximation.

Theorem 2 (Universal Approximation Theorem for multilayer FNNs)**.** *Let $\psi(\cdot)$ be any continuous nonconstant function. Then, a multilayer FNN of the form: $N(x) = \sum_{i=1}^{H} \omega_i \prod_{j=1}^{q} \psi(\omega_i^j \cdot x + b_i^j)$ is uniformly dense on the set of continuous functions C. In other words, multilayer FNNs are capable of arbitrarily accurate approximation to any real-valued continuous function over a compact set.*

For the proof of the above theorem, we will use the Stone-Weierstrass Theorem.

Theorem 3 (Stone-Weierstrass Theorem)**.** *Let A be an algebra of real continuous functions on a compact set K. If A separates points on K and if A vanishes at no point of K, then the uniform closure B of A consists of all real continuous functions on K (i.e., A is ρ_K-dense in the space of real continuous functions on K; a subset S*

of a metric space (X,ρ) is ρ -dense in a subset T if for every $\varepsilon > O$ and for every $t \in T$ there is an $s \in S$ such that $\rho(s,t) < \varepsilon$.).

Proof of Universal Approximation theorem for multilayer FNNs We apply the Stone-Weierstrass Theorem. Let K be any compact set and let A' be the set of functions of the form $\omega \cdot x + b$. Any $\psi(\cdot)$ a multilayer FNN $N(x)$ of the above form is obviously an algebra on K. If $x, y \in K, x \neq y$, then there exist a function $A \in A'$ such that $\psi(A(x)) \neq \psi(A(y))$.

To see this, pick $\alpha, \beta \in \mathbb{R}, \alpha \neq \beta$ such that $\psi(\alpha) \neq \psi(\beta)$. Pick $A(\cdot)$ to satisfy $A(x) = \alpha, A(y) = \beta$, then $\psi(A(x)) \neq \psi(A(y))$. This ensures that $N(\psi(\cdot))$ is separating on K.

Second, there exist $\psi(A(\cdot))'$s that are constant and not equal to zero. To see this, pick $\beta \in \mathbb{R}$ such that $\psi(\beta) \neq 0$ and set $A(x) = 0 \cdot x + \beta$. For all $x \in K$, $\psi(A(x)) = \psi(\beta)$. This ensures that $N(\psi(\cdot))$ vanishes at no point of K.

The Stone-Weierstrass Theorem thus implies that $N(\psi(\cdot))$ is ρ_K-dense in the space of real continuous functions on K. Because K is arbitrary .the result follows. □

What can be summarized here is that a FNN is capable of approximating uniformly any (piecewise-)continuous (multivariate) function, to any desired accuracy. This implies that any failure of a function mapping by a (multilayer) network must arise from an inadequate choice of weights and biases or an insufficient number of hidden nodes. Moreover, in the univariate case only one hidden layer is needed.

4.2 Physics Informed Random Projection Neural Networks (PINNs)

The training of an ANN requires the minimization of a cost function as in Eq. 5.2. But, even for the simple case of single layer FNNs (SLFFNs), this task may become challenging. Many optimization algorithms used for training ANNs apply stochastic gradient-based approaches which back-propagate the error and adjust the weights through specific directions [169]. More recently, second-order stochastic optimization methods have been widely investigated to get better performances than first-order methods, especially when ill-conditioned problems must be solved (see

e.g. [170] and the references therein). Nevertheless, there are still difficulties in using these approaches, such as the setting of the so-called hyperparameters, the significant increase of computing time when the number of data or the number of nodes in the hidden layer grows, and the high non-convexity stemming from the use of nonlinear activation functions, which can lead the algorithms to local minima.

A way to deal with the "curse of dimensionality" in training ANNs is to apply the concept of random projection. The idea behind random projections is to construct a Lipschitz mapping that projects the data into a random subspace. The feasibility of this approach can been justified by the celebrated Johnson and Lindenstrauss (JL) Theorem [106]:

Theorem 4 (Johnson and Lindenstrauss)**.** *Let $\mathscr{X}$ be a set of n points in $\mathbb{R}^d$. Then, $\forall\,\varepsilon \in (0,1)$ and $k \in \mathbb{N}$ such that $k \geq O(\frac{\ln n}{\varepsilon^2})$, there exists a map $F : \mathbb{R}^d \to \mathbb{R}^k$ such that*

$$(1-\varepsilon)\|u-v\|^2 \leq \|F(u)-F(v)\|^2 \leq (1+\varepsilon)\|u-v\|^2 \quad \forall\, u,v \in \mathscr{X}. \tag{4.4}$$

Note that while the above theorem is deterministic, its proof relies on probabilistic techniques combined with Kirszbraun's theorem to yield a so-called extension mapping [106]. In particular, it can be shown that one of the many such embedding maps is simply a linear projection matrix with suitable random entries. Then, the JL Theorem may be proved using the following lemma.

Lemma 1. *Let $\mathscr{X}$ be a set of n points in $\mathbb{R}^d$, $\varepsilon \in (0,1)$ and $F(u)$ the random projection defined by*

$$F(u) = \frac{1}{\sqrt{k}}, Ru, \quad u \in \mathbb{R}^d,$$

where $R = [r_{ij}] \in \mathbb{R}^{k\times d}$ has components that are i.i.d. random variables sampled from a normal distribution. Then, $\forall\, u \in \mathscr{X}$

$$(1-\varepsilon)\|u\|^2 \leq \|F(u)\|^2 \leq (1+\varepsilon)\|u\|^2$$

is true with probability $p \geq 1 - 2\exp\left(-(\varepsilon^2-\varepsilon^3)\frac{k}{4}\right)$.

Similar proofs have been given for distributions different from the normal one (see e.g. [171, 172]). In general, the proof of the JL Theorem is based on the fact that inequality (4.4) is true with probability 1 if k is large enough. Thus, the

theorem states that there exists a projection (referred to as encoder) of $\mathscr{X}$ into a random subspace of dimension $k \geq O(\frac{\ln n}{\varepsilon^2})$, where the distance between any pair of points in the embedded space $F(\mathscr{X})$ is bounded in the interval $[1-\varepsilon, 1+\varepsilon]$. For completeness, we will present the proof of the JL theorem.

Proof of JL theorem. The proof is constructive and is an example of the probabilistic method. Choose an F which is a random projection. Let $F = \frac{1}{\sqrt{k}} R||u||^2$, where R is a $k \times d$ matrix where each entry is sampled i.i.d. from a Gaussian $N(0,1)$. Note that there are $O(n^2)$ pairs of $u, v \in \mathscr{X}$. By the union bound,

$$
\begin{aligned}
&Pr(\exists\, u,v, \quad s.t. \quad the \quad following \quad fails : (1-\varepsilon)\|u-v\|^2 \leq \|\frac{1}{\sqrt{k}} R(u-v)\|^2 \leq (1+\varepsilon)\|u-v\|^2) \\
&\leq \sum_{u,v \in \mathscr{X}} Pr(s.t. \quad the \quad following \quad fails : (1-\varepsilon)\|u-v\|^2 \leq \|\frac{1}{\sqrt{k}} R(u-v)\|^2 \leq (1+\varepsilon)\|u-v\|^2) \\
&\leq 2n^2 e^{-(\varepsilon^2-\varepsilon^3)k/4} \\
&\leq 1
\end{aligned}
$$

the last step follows if we choose $k = \frac{20}{\varepsilon^2} \log n$. Note that that the probability of finding a map F which satisfies the desired conditions is strictly greater than 0, so such a map must exist. □

The proof of the norm preservation lemma has been shown in [171]. For completeness, in this step we will present the proof.

Proof of Lemma 1. First let us show that for any $u \in \mathbb{R}^d$, we have that:

$$\mathbb{E}[\|\frac{1}{\sqrt{k}} Ru\|^2] = \mathbb{E}[\|u\|^2].$$

To see this, let us examine the expected value of the entry $[Ru]_j^2$:

$$\begin{aligned}\mathbb{E}[[Ru]_j^2] &= \mathbb{E}[(\sum_{i=1}^{d} R_{i,j}u_i)^2] \\ &= \mathbb{E}[\sum_{i,i'} R_{i,j}R_{i',j}u_{i'}u_i] \\ &= \mathbb{E}[\sum_{i} R_{i,i}^2 u_i^2] \\ &= \sum_i u_i^2 \\ &= \|u\|^2\end{aligned}$$

and note that:

$$\|\frac{1}{\sqrt{k}}Ru\|^2 = \frac{1}{k}\sum_{j=1}^{k}[Ru_j^2]$$

which proves the first claim (note that all we require for this proof is independence and unit variance in constructing R). Note that above shows that $\tilde{Z}_j = [Ru]_j/\|u\|$ is distributed as $N(0,1)$, and $\tilde{Z}_j$ are independent. We now bound the failure probability of one side. By the union bound,

$$\begin{aligned}Pr(\|\frac{1}{\sqrt{k}}Ru\|^2 > (1+\varepsilon)\|u\|^2) &= Pr(\sum_{i=1}^{k}\tilde{Z}_i^2 > (1+\varepsilon)k) \\ &= n^2 Pr(\chi_k^2 > (1+\varepsilon)k)\end{aligned}$$

(where χ_k^2 is the chi-squared distribution with k degrees of freedom). Now we appeal to a concentration result below, which bounds this probability by:

$$\leq exp(-\frac{k}{4}(\varepsilon^2 - \varepsilon^3).$$

A similar argument handles the other side (and the factor of 2 in the bound). □

The following lemma for χ^2-distribution [173] was used in the above proof

Lemma 2. *We have that:*

$$\begin{aligned}Pr(\chi_k^2 \geq (1+\varepsilon)k) &\leq exp(-\frac{k}{4})(\varepsilon^2 - \varepsilon^3) \\ Pr(\chi_k^2 \leq (1-\varepsilon)k) &\leq exp(-\frac{k}{4})(\varepsilon^2 - \varepsilon^3).\end{aligned}$$

Proof Let $Z_1, Z_2, \ldots, Z_k$ be i.i.d. $N(0,1)$ random variables. By Markov's inequality:

$$\begin{aligned} Pr(\chi_k^2 \geq (1+\varepsilon)k) &= Pr(\sum_{i=1}^{k} Z_i^2 > (1+\varepsilon)k) \\ &= Pr(e^{\lambda \sum_{i=1}^{k} Z_i^2} > e^{(1+\varepsilon)k\lambda}) \\ &\leq \frac{\mathbb{E}[e^{\lambda \sum_{i=1}^{k} Z_i^2}]}{e^{(1+\varepsilon)k\lambda}} \\ &= \frac{(\mathbb{E}[e^{\lambda Z_1^2}])^k}{e^{(1+\varepsilon)k\lambda}} \\ &= e^{(1+\varepsilon)k\lambda}(\frac{1}{1-2\lambda})^{\frac{k}{2}} \end{aligned}$$

where the last step follows from evaluating the expectation, which holds for $0 < \lambda \leq 1/2$ (this expectation is just the moment generating function). Choosing $\lambda = \frac{\varepsilon}{2(1+\varepsilon)}$ which minimizes the above expression (and is less than $1/2$ as required), we have:

$$\begin{aligned} Pr(\chi_k^2 \geq (1+\varepsilon)k) &= ((1+\varepsilon)e^{-\varepsilon})^{\frac{k}{2}} \\ &\leq exp(-\frac{k}{4}(\varepsilon^2 - \varepsilon^3)). \end{aligned}$$

using the upper bound $1+\varepsilon \leq exp(\varepsilon - (\varepsilon^2 - \varepsilon^3)/2)$. The other bound is proved in a similar manner. □

Moreover, in [171] it was proved that if the random projection is of Gaussian type, then a lower bound of the embedding dimension is given by $k \geq 4(\varepsilon^2/2 - \varepsilon^3/3)^{-1} \ln n$. We note that the above mapping is a feature mapping, which in principle may result in a dimensionality reduction ($k < d$) or a projection into a higher-dimensional space ($k > d$) in which one seeks a linear manifold (in analogy to the case of kernel-based manifold learning methods). We also note that while the above linear random projection is but one of the choices for constructing a JL embedding (and proving it), it was experimentally demonstrated and/or theoretically proven that appropriately constructed nonlinear random embeddings may outperform simple linear random projections (see e.g. [174, 175, 103–105, 176, 97, 100, 177]).

Furthermore, within this framework, it has been shown that single-layer FNNs with randomly assigned input weights and biases of the hidden layer and with infinitely differentiable functions at the hidden layer, called ELMs, can universally approximate any continuous function on any compact input set [176, 103–105]. For

the case of an FNN with a single hidden layer of h units, the random projection of the input space can be written as

$$Y = \Phi(X), \tag{4.5}$$

where the columns of the matrix $X \in \mathbb{R}^{d\times n}$ represent a set of n points in the input d-dimensional space, the columns of $Y \in \mathbb{R}^{k\times n}$ are the corresponding random projections, and $\Phi : \mathbb{R}^{d\times n} \to \mathbb{R}^{k\times n}$ acts as an encoder, i.e. a family of transfer functions whose parameters are sampled from a certain random distribution function. If the values of the weights w_{ij} between the input and the hidden layer are fixed, then $\forall x \in \mathbb{R}^d$ the random projection can be written as a linear map:

$$Y = W^o\Phi, \quad Y \in \mathbb{R}^{k\times n}, \tag{4.6}$$

where $\Phi \in \mathbb{R}^{h\times n}$ is a random matrix containing the outputs of the nodes of the hidden layer as shaped by the h random distribution functions (e.g. RBFs or sigmoidal functions) and the d-dimensional inputs.

Thus, the so-called ELMs can be seen as underdetermined linear systems in which the output weights are estimated by solving minimum-norm least squares problems

Theorem 5 ([176])**.** *Let us consider a single-hidden-layer FNN with h hidden units and an infinitely differentiable transfer function $\phi : \mathbb{R} \to \mathbb{R}$, n distinct input-output pairs $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}^k$, and randomly chosen values from any continuous probability distribution for the internal weights w_{ij} and for the values of the biases of the h neurons of the hidden layer, grouped in $b \in \mathbb{R}^h$. Let us also denote by $W \in \mathbb{R}^{h\times d}$ and $W^o \in \mathbb{R}^{k\times h}$ the matrices containing the internal and output weights, and by $X \in \mathbb{R}^{d\times n}$ and $Y \in \mathbb{R}^{k\times n}$ the matrices with columns x_i and y_i. Then, the hidden layer output matrix $\Phi \in \mathbb{R}^{h\times n}$, whose elements are determined by the action of the transfer functions on $WX + b1^T$ (where $1 \in \mathbb{R}^n$ has all the entries equal to 1), has full rank and*

$$\|W^o\Phi - Y\| = 0$$

with probability 1.

A review on neural networks with random weights can be found in [178].

4.3 Diffusion Maps

Diffusion Maps is a non-linear manifold learning algorithm ([79, 6, 80]) that identifies a low-dimensional representation $y_i \in \mathbb{R}^{\mu}$ of a point $z_i \in \mathbb{R}^n, i = 1,2,\ldots N$ in the high dimensional space ($\mu << n$) addressing the diffusion distance among points as the preserved metric ([80]). Difussion Maps assume that the data lie on a smooth manifold, while it can be shown that the eigenvectors of the large normalized kernel matrices constructed from data converge to the eigenfunctions of Laplace-Beltrami operator on this manifold at the limit of infinite data ([79, 80]). The approximation of this Laplace-Betrami operator is made by representing the weighted edges connecting nodes i and j commonly by a normalized diffusion kernel matrix W with elements:

$$w_{ij} = \exp\left(-\frac{||z_i - z_j||^2}{\sigma}\right), \tag{4.7}$$

Then, one can define the $N \times N$ diffusion matrix P by:

$$P = D^{-1}W, D = \text{diag}\left(\sum_{j=1}^{N} w_{ij}\right) \tag{4.8}$$

whose elements p_{ij} correspond to the probability of jumping from one point to another in the high-dimensional space.

Taking the power t of the diffusion matrix P is essentially identical of observing t steps forward of a Markov chain process Z_t on the data points and the element $P^t(z_i, z_j)$ denotes the transition probability of moving from point i to point j after t steps by setting:

$$p_{ij} = p(z_i, z_j) = Prob(Z_{t+1} = z_j | Z_t = z_i). \tag{4.9}$$

The random walk on this weighted graph can be defined by the below transition probabilities:

$$p_{ij} = p(z_i, z_j) = \frac{w_{ij}}{deg(z_i)}, \tag{4.10}$$

where $deg(z_i)$ denotes the weighted degree of the point i, defined as:

$$deg(z_i) = \sum_j w_{ij}. \tag{4.11}$$

At the next step it is easy to compute the so called graph Laplacian matrix $\tilde{P}$:

$$\tilde{P} = \tilde{D}^{1/2} P D^{-1/2}, \tag{4.12}$$

The eigendecomposition of $\tilde{P}$ results to $\tilde{P} = U\Lambda U^*$, where Λ is a diagonal matrix with the eigenvalues and U is the matrix with columns the eigenvectors of $\tilde{P}$. The eigenvalues of P are the same of $\tilde{P}$ since P is the adjoined of the symmetric matrix P while the left and right eigenvectors of P (say Φ and Ψ) are related to those of $\tilde{P}$ as ([82]):

$$\Phi = UD^{1/2}, \qquad \Psi = UD^{-1/2}. \tag{4.13}$$

The embedding of the manifold in μ dimensions consists of the first μ non-trivial/dependent eigenvectors of $\tilde{P}$:

$$y_i = (\lambda_1^t \phi_{1,i}, \ldots, \lambda_\mu^t \phi_{\mu,i}), \qquad i = 1, \ldots, N, \tag{4.14}$$

where t denotes the number of diffusion steps (usually $t = 0$) and $\lambda_1, \ldots, \lambda_\mu$ the descending order eigenvalues. The diffusion distance, for two data points z_i and z_j at the time step t is defined as:

$$D_t^2(z_i, z_j) = \sum_k \frac{(p_t(z_i, z_k) - p_t(z_j, z_k))^2}{\Phi_0(z_k)}, \tag{4.15}$$

where Φ_0 denotes the stationary distribution of the random walk described by the diffusion matrix P [179]:

$$\Phi_0(y_i) = \frac{deg(z_i)}{\sum_{z_j \in Y} deg(z_j)}. \tag{4.16}$$

In practice, the embedding dimension μ is determined by the spectral gap in the eigenvalues of the final decomposition. Such a numerical gap means that the first few eigenvalues would be adequate for the approximation of the diffusion distance between all pairs of points [79, 6]. Here we retain only the μ parsimonious eigendimensions of the final decomposition as proposed in [180, 181].

One very important feature of the Diffusion Maps is that preserves the intrinsic geometry of the data. The below theorem provide the appropriate theoretical validation of the Diffusion Maps algorithm.

Theorem 6. *If we choose our diffusion coordinates as in 4.14, then the diffusion distance between points in data space (measured using the D^{-1} metric) is equal to the Eucliden distance in the diffusion space.*

The above theorem has been proved in [80]. We will present the proof for completeness purposes:

Proof. We are required to prove that

$$D_t(z_i,z_j)^2 = ||p_t(z_i,\cdot) - p_t(z_j,\cdot)||^2_{l^2(\mathbb{R}^\mu,D^{-1})} = \sum_k \lambda_k^{2t}(\phi_k[i]-\phi_k[j])^2.$$

Here, $p_t(z_i,z_j) = P_{ij}$ are the probabilities which form the componetns of the diffusion matrix. Let assume for simplicity that $t = 1$. Then:

$$D(z_i,z_j)^2 = ||p(z_i,\cdot) - p(z_j,\cdot)||^2_{l^2(\mathbb{R}^\mu,D^{-1})} = ||P[i,\cdot] - P[j,\cdot]||^2_{l^2(\mathbb{R}^\mu,D^{-1})}.$$

According to the eigendecomposition of the equation 4.12 $P = \sum_k \lambda_k \phi_k e_k^T$, this equals

$$\begin{aligned}
&|\sum_k \lambda_k \phi_k[i] e_k^T - \sum_{k\geq 0} \lambda_k \phi_k[i] e_k^T|^2 \\
&\quad = |\sum_k \lambda_k e_k^t (\phi_k[i] - \phi_k[j])|^2 \\
&= |\sum_k \lambda_k y_k'^T D^{\frac{1}{2}} (\phi_k[i] - \phi_k[j])|^2 \\
&= |\sum_k \lambda_k y_k'^T (\phi_k[i] - \phi_k[j]) D^{\frac{1}{2}}|^2.
\end{aligned}$$

In $l^2(\mathbb{R}^\mu, D^{-1})$, this distance is:

$$\begin{aligned}
&(\sum_k \lambda_k y_k'^T (\phi_k[i] - \phi_k[j]) D^{\frac{1}{2}}) D^{-1} (\sum_m \lambda_m y_m'^T (\phi_m[i] - \phi_m[j]) D^{\frac{1}{2}})^T \\
&= (\sum_k \lambda_k y_k'^T (\phi_k[i] - \phi_k[j]) D^{\frac{1}{2}}) D^{-1} (D^{\frac{1}{2}} \sum_m \lambda_m y_m^T (\phi_m[i] - \phi_m[j])) \\
&\quad = \sum_k \lambda_k y_k'^T (\phi_k[i] - \phi_k[j]) \sum_m \lambda_m y_m^T (\phi_m[i] - \phi_m[j]).
\end{aligned}$$

Since, $\{y_k'\}$ is an orthonormal set

$$y_m'^T y_k' = 0, \quad for \quad m \neq k.$$

Therefore,

$$D(z_i, z_j)^2 = \sum_k \lambda_k^2 (\phi_k[i] - \phi_k[j])^2. \quad \square$$

Chapter 5

Numerical Solution of Stiff ODEs with Physics-Informed Random Projection Neural Networks.

5.1 FNNs for the solution of differential Equations

We aim to solve IVPs of stiff ODEs of the following form:

$$\begin{cases} \dfrac{dy_i}{dx} &= f_i(x, y_1, y_2, ..., y_m), \\ y_i(x_0) &= \alpha_i, \end{cases} \qquad i = 1, 2, \ldots, m, \tag{5.1}$$

where the functions f_i and the initial values α_i are known, and the functions $y_i(x)$ are the unknowns. In order to simplify the notation, we group the functions $y_i(x)$ in a vector function $y(x) : \mathbb{R} \to \mathbb{R}^m$, and the functions $f_i(x, y_1, y_2, ..., y_m)$ in $f(x, y)$. Such systems may also arise after discretization in space of PDEs.

Given a set of n input points $x_j \in \mathbb{R}^d$, defining the grid where the solution is sought the "classical way" to solve differential equations in a d-dimensional domain with FNNs involves the solution of a minimization problem of the form:

$$\min_P E(P) := \sum_{j=1}^{n} \left\| \frac{d\Psi}{dx}(x_j, P, q) - f(x_j, \Psi(x_j, P, q)) \right\|^2. \tag{5.2}$$

$\Psi(x,P,q) = \Omega(x,N(x,P,q))$ defines a trial solution with m components, each associated with a component y_i of the solution y, $\Omega : \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^m$ is sufficiently smooth, $N(x) = N(x,P,q)$ is a FNN with m components $N_i(x,p_i,q)$, and P is a matrix containing the network parameters (the elements of the matrix containing the weights from the hidden layer to the output layer $W^o = [w^o_{jl}] \in \mathbb{R}^{m\times h}$, the elements of the matrix containing the weights from the input to the hidden layer $W = [w_{ij}] \in \mathbb{R}^{h\times d}$, the elements of the vector of the biases of the hidden nodes $b \in \mathbb{R}^h$; the vector $q \in \mathbb{R}^s$ contains the hyperparameters of the neural network, such as the parameters of the vector with the activation functions $\Phi : \mathbb{R}^h \times \mathbb{R}^s \to \mathbb{R}^h$ with components the h activation functions Φ_j (e.g. for radial basis functions, the biases of the hidden neurons and the variances of the Gaussian functions), the learning rate, and the batch size. In the case of ODEs, according to the above notation $d = 1$, i.e. the input domain is one-dimensional (representing for example time). In order to solve the optimization problem (5.2), one usually needs quantities such as the derivatives of $N_i(x,p_i,q)$ with respect to the input x and the weights and biases represented by p_i. These can be obtained in several ways, e.g. by computing analytical derivatives, by using finite difference or other numerical approximations, by symbolic differentiation or by automatic differentiation (see e.g. [182] and the references therein).

5.2 The proposed method

Here we focus on the numerical solution of problem (5.1) in an interval, say $[x_0, x_{end}]$. According to the previous notation, for this problem we have $d = 1$. Thus, we denote by $\Psi_i(x, w^o_i, p_i)$ the i-th component of the trial solution $\Psi(x, W^o, P)$, where $W^o \in \mathbb{R}^{m\times h}$ is the matrix containing the weights w^o_{ij} between the hidden and the output layer. Note that we separate W^0 from the other network parameters consisting of the input weights, the biases of the hidden layer and the parameters of the transfer function, and denote those parameters again by P, with a small abuse of the notation (thus integrating to it the vector of the hyperparameters q).

Following [8] and taking into account that the trial solution must satisfy the initial value conditions of the problem, i.e. $y_i(x_0) = \alpha_i$, $i = 1,2,\ldots,m$, we set

$$\Psi_i(x, w^o_i, p_i) = \alpha_i + (x - x_0)N_i(x, w^o_i, p_i), \tag{5.3}$$

where $N_i(x, w_i^o, p_i)$ is a single-output FNN with parameters the output weights $w_i^o = [w_{1i}^o \; w_{2i}^o \; \ldots \; w_{hi}^o]^T \in \mathbb{R}^h$, and p_i contains the remaining parameters associated with that network.

Then, if one considers the numerical solution based on n collocation points $x_1, x_2, \ldots, x_n$, then the error function that we seek to minimize for training the FNN is given by

$$E(W^o, P) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(\frac{d\Psi_i}{dx}(x_j, w_i^o, p_i) - f_i(x_j, \Psi_1(x_j, w_1^o, p_1), \ldots, \Psi_m(x_j, w_m^o, p_m) \right)^2. \tag{5.4}$$

Here we propose a machine learning method based on random projections for the solution of IVPs of ODEs in n input collocation points. In particular, we employ m (one for each unknown variable of the systems of ODEs) SISO neural sub-networks with a linear output transfer function, with a single hidden layer having h nodes with Gaussian RBFs.

In particular, we consider each sub-network N_i to be a linear combination of RBFs,

$$N_i(x, w_i^o, p_i) = \sum_{j=1}^{h} w_{ji}^o G_{ji}(x), \quad i = 1, 2, \ldots, m, \tag{5.5}$$

where

$$G_{ji}(x) = G(w_{ji}x + b_{ji}, c_j, \sigma_{ji}) = \exp\left(-\frac{(w_{ji}x + b_{ji} - c_j)^2}{\sigma_{ji}^2} \right), \tag{5.6}$$
$$j = 1, 2, \ldots h, \; i = 1, 2, \ldots m,$$

with $w_{ji} = 1$. The hyperparameters corresponding to the centers of the RBFs are set to $c_j = x_0 + (j-1)\bar{s}$ with $\bar{s} = (x_{end} - x_0)/(h-1)$ and $j = 1, \ldots, h$.

Under the above assumptions, the derivative of the i-th component Ψ_i of the trial solution with respect to the collocation point x_l is given by:

$$\frac{\partial \Psi_i}{\partial x_l} = N_i(x_l, w_i^o, p_i) - (x_l - x_0) \sum_{j=1}^{h} \frac{2}{\sigma_{ji}^2} w_{ji}^o (x_l + b_{ji} - c_j) \exp\left(-\frac{(x_l + b_{ji} - c_j)^2}{\sigma_{ji}^2} \right), \tag{5.7}$$

while, for any fixed x_l, the derivative of Ψ_i with respect to the only unknown parameter w^o_{ji} is given by:

$$\frac{\partial \Psi_i}{\partial w^o_{ji}} = (x_l - x_0) \exp\left(-\frac{(x_l + b_{ji} - c_{ji})^2}{\sigma^2_{ji}} \right). \tag{5.8}$$

For the determination of the other two sets of hyperparameters of the Gaussian RBFs, namely b_{ij} and σ_{ji}, we fix reference intervals in which we assign random values for these parameters. Our requests are that the functions are neither too steep nor too flat in the reference interval, and at each collocation point there are at least two basis functions giving values that are not too small. Then, based on numerical experiments, the biases b_{ji} of the hidden units and the parameters $1/\sigma^2_{ji}$ are taken to be uniformly randomly distributed in the intervals

$$\left[-\frac{(x_{end} - x_0)}{6}, 0 \right] \text{ and } \left[\frac{3}{8(x_{end} - x_0)^2}, \frac{81}{2(x_{end} - x_0)^2} \right],$$

respectively. We emphasise that this choice appears to be problem independent. Therefore, the only parameters that have to be determined by training the network are the output weights w^o_{ji}. Hence, for the n collocation points x_l, the outputs of each network N_i, $i = 1, 2, \ldots m$, are given by:

$$N_i(x_1, x_2, \ldots x_n, w^o_i, p_i) = R_i w^o_i, \tag{5.9}$$

where $N_i(x_1, x_2, \ldots x_n, w^o_i, p_i) \in \mathbb{R}^n$ is the vector with l-th component the output of N_i corresponding to x_l, and $R_i = R_i(x_1, \ldots, x_n, p_i) \in \mathbb{R}^{n \times h}$ is defined as

$$R_i(x_1, \ldots, x_n, p_i) = \begin{bmatrix} G_{1i}(x_1) & \cdots & G_{hi}(x_1) \\ \vdots & \vdots & \vdots \\ G_{1i}(x_n) & \cdots & G_{hi}(x_n) \end{bmatrix}. \tag{5.10}$$

The minimization of the error function given in (5.4) is performed by a Gauss-Newton scheme (see e.g. [183]) over nm nonlinear residuals F_q, with $q = l + (i-1)n$, $i = 1, 2, \ldots m$, $l = 1, 2, \ldots n$, given by

$$F_q(W^o) = \frac{d\Psi_i}{dx_l}(x_l, w^o_i, p_i) - f_i(x_l, \Psi_1(x_l, w^o_1, p_1), \ldots, \Psi_m(x_l, w^o_m, p_m)), \tag{5.11}$$

where $W^0 \in \mathbb{R}^{mh}$ is the column vector obtained by stacking the values of all the m vectors $w_i^o \in \mathbb{R}^h$, $W^o = [W_{k=1,2,\dots mh}] = [w_1^o, w_2^o \dots, w_m^o]^T$.
Thus, by setting $F(W^o) = [F_1(W^o) \cdots F_q(W^o) \cdots F_{(nm)}(W^o)]^T$, the new update $dW^{o(\nu)}$ at the (ν)-th iteration is computed by the solution of the (under-determined) system

$$\nabla_{W^{o(\nu)}} F \, dW^{o(\nu)} = -F(W^{o(\nu)}), \tag{5.12}$$

where $\nabla_{W^{o(\nu)}} F \in \mathbb{R}^{nm \times mh}$ is the Jacobian matrix of F with respect to $W^{o(\nu)}$:

$$\nabla_{W^{o(\nu)}} F(W^{o(\nu)}) = \begin{bmatrix} \frac{\partial F_1}{\partial W_1^o} & \frac{\partial F_1}{\partial W_2^o} & \cdots & \frac{\partial F_1}{\partial W_p^o} & \cdots & \frac{\partial F_1}{\partial W_{mh}^o} \\ \frac{\partial F_2}{\partial W_1^o} & \frac{\partial F_2}{\partial W_2^o} & \cdots & \frac{\partial F_2}{\partial W_p^o} & \cdots & \frac{\partial F_2}{\partial W_{mh}^o} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial F_q}{\partial W_1^o} & \frac{\partial F_q}{\partial W_2^o} & \cdots & \frac{\partial F_q}{\partial W_p^o} & \cdots & \frac{\partial F_q}{\partial W_{mh}^o} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial F_{nm}}{\partial W_1^o} & \frac{\partial F_{nm}}{\partial W_2^o} & \cdots & \frac{\partial F_{(nm)}}{\partial W_p^o} & \cdots & \frac{\partial F_{nm}}{\partial W_{mh}^o} \end{bmatrix}_{|(W^{o(\nu)})} . \tag{5.13}$$

Note that the residuals depend on the derivatives $\frac{\partial \Psi_i}{\partial x_l}$ in (5.7) and the trial functions Ψ_i in (5.3), while the elements of the Jacobian matrix depend on the derivatives of $\frac{\partial \Psi_i}{\partial w_{ji}^o}$ in (5.8) as well as on the mixed derivatives $\frac{\partial^2 \Psi_i}{\partial x_l \partial w_{ji}^o}$. Based on (5.7), the latter are given by

$$\frac{\partial^2 \Psi_i}{\partial x_l \partial w_{ji}^o} = \frac{\partial N_i(x_l, w_i^o, p_i)}{\partial w_{ji}^o} - (x_l - x_0)\frac{2}{\sigma_{ji}^2}(x_l + b_{ji} - c_j)\exp\left(-\frac{(x_l + b_{ji} - c_j)^2}{\sigma_{ji}^2}\right), \tag{5.14}$$

where

$$\frac{\partial N_i(x_l, w_i^o, p_i)}{\partial w_{ji}^o} = \exp\left(-\frac{(x_l + b_{ji} - c_j)^2}{\sigma_{ji}^2}\right). \tag{5.15}$$

Note also that if the Jacobian of system (5.1), with elements $\frac{\partial f_i(x_l)}{\partial y_k}$, is given, then the elements of the Jacobian matrix $\nabla_{W^{o(\nu)}} F$ can be computed easily:

$$\frac{\partial F_p}{\partial W_q^o} = \frac{\partial^2 y_i}{\partial x_l \partial w_{jk}^o} - \frac{\partial f_i(x_l)}{\partial w_{jk}^o} = \frac{\partial^2 \Psi_i}{\partial x_l \partial w_{jk}^o} - (x - x_0)\frac{\partial f_i(x_l)}{\partial y_k}\frac{\partial N_k(x_l, w_i^o, p_i)}{\partial w_{jk}^o}, \tag{5.16}$$

where, as before, $q = l + (i-1)n$ and $p = j + (k-1)h$.

As in general $h > n$, i.e. the minimization problem in (5.12) is an underdetermined linear system, we compute its solution subject to the minimum L_2 norm. This, for a full row rank Jacobian is given by:

$$dW^{o(\nu)} = -\nabla_{W^{o(\nu)}} F^T (\nabla_{W^{o(\nu)}} F^T \nabla_{W^{o(\nu)}} F) F(W^{o(\nu)}). \tag{5.17}$$

In the case where the Jacobian is rank deficient, one can compute the solution by resorting to the Singular Value Decomposition for the estimation of the pseudoinverse by cutting off all the singular values (and their corresponding singular vectors) below a small tolerance ε. Furthermore, this choice also allows us to cope with the difference between the exact rank and the numerical rank of the Jacobian matrix. Following [184], for some small ε the ε-rank of a matrix $M \in \mathbb{R}^{mn \times mh}$ is defined as follows:

$$r_\varepsilon = \min\{\text{rank}(B) \in \mathbb{R}^{mn \times mh} : \|M - B\|_{L_2} \leq \varepsilon\}. \tag{5.18}$$

Then, if ε is "small enough", we neglect all the singular values below ε and approximate the pseudoinverse of the Jacobian matrix as

$$(\nabla_{W^o} F)^+ = V_{r_\varepsilon} \Sigma_{r_\varepsilon}^+ U_{r_\varepsilon}^T, \tag{5.19}$$

where $\Sigma_{r_\varepsilon} \in \mathbb{R}^{r_\varepsilon \times r_\varepsilon}$ is the diagonal matrix with the r_ε largest singular values of $\nabla_{W^o} F$, $\Sigma_{r_\varepsilon}^+$ is its pseudoinverse, and $U_{r_\varepsilon} \in \mathbb{R}^{mn \times r_\varepsilon}$ and $V_{r_\varepsilon} \in \mathbb{R}^{mh \times r_\varepsilon}$ are the matrices with columns the corresponding r_ε left and right eigenvectors, respectively. The value of ε used in our experiments is specified at the beginning of Chapter 4. Thus, the update $dW^{o(\nu)}$ is given by

$$dW^{o(\nu)} = -V_{r_\varepsilon} \Sigma_{r_\varepsilon}^+ U_{r_\varepsilon}^T F(W^{o(\nu)}).$$

5.2.1 Convergence

We note that in the above configuration, we fix all the internal weights to 1 and the centers to be equidistant in the domain, while we set randomly, from appropriately chosen uniform distributions, the biases b_{ji} and the width parameters σ_{ji} of the RBFs. This configuration is slightly different from the classical ELMs with RBFs, where the internal weights are set equal to 1 and the biases equal to 0, while the centers are uniformly randomly distributed in $[-1, 1]$ (see e.g. [107]). However, it

is straightforward to show that in our scheme we still get universal approximation properties, as stated in the next theorem.

Theorem 7. *Let us fix a continuous (target) function φ and consider the system of functions $\{N_i(x, w_i^o, p_i)\}_{i=1,\dots,m}$ in* (5.5)*, with G_{ji}, w_{ji} and c_j defined in* (5.6) *and the subsequent lines. Then, for every sequence of randomly chosen parameters p_i there exists a choice of w_i^o such that*

$$\lim_{m\to\infty} \|N_i(x, w_i^o, p_i) - \varphi(x)\| = 0 \text{ with probability 1.}$$

Proof. Since w_{ji} and c_j are fixed, one can manipulate (5.6) and write it as

$$G_{ji}(x) = \exp\left(-\frac{(x+\alpha_{ji})^2}{\beta_{ji}}\right),$$

where the parameters $\alpha_{ji} = b_{ji} - c_j$ and $\beta_{ji} = \sigma_{ji}^2$ are random variables (because b_{ji} and σ_{ji} are random variables sampled from continuous probability distributions). Therefore, the network fits the hypotheses of an FNN with random hidden nodes. Because the considered RBFs are sufficiently regular, Theorem II.1 in [107] holds and hence the thesis follows in a straightforward manner. □

5.2.2 Algorithm

The proposed method for using RPNNs for the numerical solution of stiff systems of ODEs as a pseudo-code algorithm is shown in algorithm 1.

5.3 Case study problems

We implemented Algorithm 1 using MATLAB 2020b on an Intel Core i5-8265U CPU @ 1.60GHz with up to 3.9 GHz frequency and a memory of 8 GBs. The Moore-Penrose pseudoinverse of $\nabla_{w^o} F$ was computed with the matlab built-in function `pinv`, with the default tolerance. For our simulations, we chose three well-known and challenging stiff ODE problems: Prothero-Robinson [108], van der Pol [109] and ROBER [110]. For comparison purposes, we also solved the ODE problems with two widely-used functions of the matlab ODE suite[185], namely

Algorithm 1 solving an IVP of ODE systems using an RPNN with RBFs

Require: $\dfrac{dy_i}{dx} = f_i(x, y_1, y_2, \ldots, y_m)$ in $[x_0, x_{end}]$

Require: $y_i(x_0) = \alpha_i$, $i = 1, 2, \ldots, m$ ▷ *initial conditions*

$h \leftarrow 40$; $n \leftarrow 20$ ▷ *set # neurons and collocation points*

$maxiter \leftarrow 4$ ▷ *set max # iters for Gauss-Newton method*

Select $x_l \in [x_0, x_{end}]$, $l = 1, \ldots, n$ ▷ *set collocation points*

$c_j \leftarrow x_0 + j\dfrac{x_{end} - x_0}{h-1}$, $j = 1, \ldots, h$ ▷ *set RBF centers*

$b_{ji} \sim \mathscr{U}\left(-\dfrac{x_{end} - x_0}{6}, 0\right)$, $j = 1, \ldots, h$, $i = 1, \ldots, m$ ▷ *set biases*

$\dfrac{1}{\sigma_{ji}^2} \sim \mathscr{U}\left(\dfrac{3}{8(x_{end} - x_0)^2}, \dfrac{81}{2(x_{end} - x_0)^2}\right)$, j, i as above ▷ *set RBF inverse widths*

$N_i(x, w_i^0, p_i) \leftarrow \sum_{j=1}^{h} w_{ji}^o \exp\left(-\dfrac{(x - c_j)^2}{\sigma_{ji}^2} - b_{ji}\right)$ ▷ *see* (5.5)-(5.6)

$\Psi_i(x, w_i^0, p_i) = \alpha_i + (x - x_0) N_i(x, w_i^0, p_i)$

$iter \leftarrow 0$

repeat

  for $j = 1, \ldots, n$; $i = 1, \ldots, m$ **do**

    $q \leftarrow j + (i-1)n$

    $F_q(W^o) \leftarrow \dfrac{d\Psi_i}{dx_l}(x_l, w_i^o, p_i) - f_i(x_l, \Psi_1(x_l, w_1^o, p_1), \ldots, \Psi_m(x_l, w_m^o, p_m))$

  Set $F(W^o) = [F_1(W^o), F_2(W^o), \ldots, F_{(nm)}(W^o)]^T$

  Compute Jacobian matrix $\nabla_{W^o} F(W^o)$ ▷ *see* (5.13)-(5.16)

  $(\nabla_{W^o} F)^+ \leftarrow V_{r_\varepsilon} \Sigma_{r_\varepsilon}^+ U_{r_\varepsilon}^T$ ▷ *compute pseudo-inverse*

  $W^o \leftarrow W^o - (\nabla_{W^o} F(W^o))^\dagger F(W^o)$ ▷ *update weights*

  $err \leftarrow ||F(W^o)||_2$ ▷ *compute error*

  $iter \leftarrow iter + 1$

until $(err \leq tol)$ or $(iter \geq maxiter)$

`ode45`, implementing an adaptive-step Runge-Kutta method based on the Dormand-Prince pair, and `ode15s`, implementing a variable-step variable-order multistep method based on numerical differentiation formulas. In order to estimate the error in the approximate solution, we used as reference solution the one computed by `ode15s` with absolute and relative error tolerances equal to 1e−14. To this aim, we computed the L_2 and L_∞ norms of the differences between the computed and the reference solutions, as well as the Mean Absolute Error (MAE). These performance metrics were evaluated at equidistant collocation points, selecting their number according to the problem (see below). Finally, we ran each solver 10 times and computed the median, maximum and minimum execution times in seconds. The time of each run was measured by using the matlab commands `tic` and `toc`.

Henceforth, we use t instead of x, since the independent variable in the test problems represents time. We also assume $t_0 = x_0 = 0$.

5.3.1 Case study 1: Prothero-Robinson problem

The Prothero-Robinson ODE [108] is given by

$$\frac{dy}{dt} = \lambda(y - \phi(t)) + \phi'(t), \quad \lambda < 0. \tag{5.20}$$

Its solution is $\phi(t)$ and the initial condition is $y(0) = \phi(0)$. The problem becomes stiff for $\lambda \ll 0$. For our numerical simulations, we choose $\phi(t) = sin(t)$, $u(0) = \phi(0) = sin(0) = 0$, and $[0, 2\pi]$ as the time interval where the solution is sought, while the parameter λ controlling the stiffness is set equal to -30. For the implementation of the proposed approach, we use the following (initial) trial solution:

$$\Psi(t, w^o) = \alpha^{(0)} + tN(t, w^o), \quad \alpha^{(0)} = y(0) = sin(0) = 0. \tag{5.21}$$

In Figure 5.1, we show the approximate solutions obtained with tolerances 1e−02 and 1e−03.

In Table 5.1, we report the numerical approximation accuracy obtained with the various methods in terms of the L_2-norm and L_∞-norm of the error and of MAE with respect to the reference solution. As it is shown, the proposed numerical scheme outperforms both `ode45` and `ode15s` in all metrics for both tolerances.

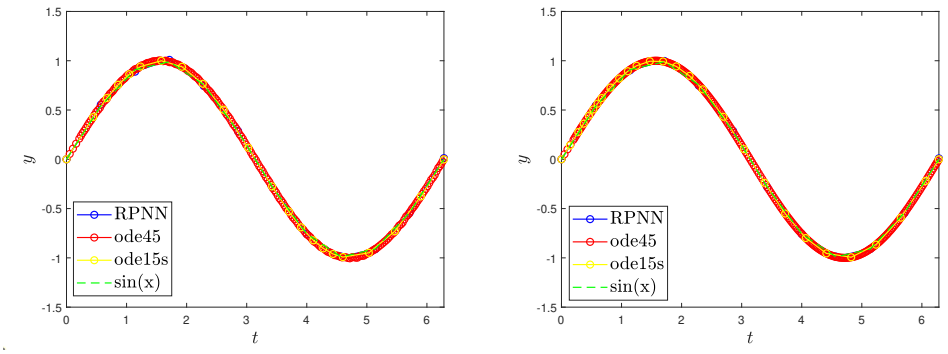


Fig. 5.1 Numerical solutions of the Prothero-Robinson problem with $\lambda = -30$ in the interval $[0, 2\pi]$: (a) $tol = 1e{-}02$, (b) $tol = 1e{-}03$. The reference (analytical) solution is $sin(t)$.

Table 5.1 Prothero-Robinson problem with $\lambda = -30$ in the interval $[0, 2\pi]$. Absolute errors (L_2-norm, L_∞-norm and MAE) for the solutions computed with tolerances $1e{-}02$ and $1e{-}03$. The reference solution is the analytical solution $sin(t)$.

	$tol = 1e{-}02$			$tol = 1e{-}03$		
	L_2	L_∞	MAE	L_2	L_∞	MAE
RPNN	5.95e−08	6.12e−09	3.31e-10	2.23e−08	2.84e−09	9.36e-11
`ode45`	2.24e−02	1.13e−03	2.97e−04	2.19e−05	1.14e−06	2.89e−07
`ode15s`	7.36e−03	5.86e−04	1.01e−04	5.65e−06	6.14e−07	6.08e−08

Table 5.2 Prothero-Robinson problem with $\lambda = -30$ in the interval $[0, 2\pi]$. Computational times in seconds (median, $5thpercetnile$ and $95thpercentile$) and number of points required with tolerances $1e{-}02$ and $1e{-}03$. The reference solution is the analytical solution $sin(t)$

	$tol = 1e{-}02$				$tol = 1e{-}03$			
	median	$5thperc$	$95thperc$	# pts	median	$5thperc$	$95thperc$	# pts
RPNN	7.87e−04	5.65.e−04	1.23e−03	12	6.67e−04	5.11e−04	6.35e−03	12
`ode45`	4.21e−01	4.10e−01	5.12e−01	189173	4.88e−01	4.45e−01	5.02e−01	189201
`ode15s`	7.65e−04	7.19e−04	1.05e−03	19	1.12e−03	1.01e−03	1.54e−03	65

Finally, in Table 5.2 we report the computational times and the number of points required by each method, for both tolerance values. Notably, our method outperforms both `ode45` and `ode15s`, since it results in significantly smaller computational times and number of points than `ode45`, and (on average) in slightly smaller computational times and number of points than `ode15s`. We note that `ode45` has a "poor" performance due to the very high stiffness, thus requiring a huge number of points and yet resulting in relatively large approximations errors.

5.3.2 Case study 2: van der Pol problem

The van der Pol model and the concept of the so-called relaxation oscillations was introduced by Balthazar van der Pol [109] to describe the nonlinear oscillations observed in a triode circuit characterized by a slow charge of a capacitor followed by a very rapid discharge. The model is given by the following equations:

$$\begin{aligned} y_1' &= y_2, \\ y_2' &= \mu(1-y_1^2)y_2 - y_1. \end{aligned} \tag{5.22}$$

The problem becomes stiff for $\mu \gg 1$. As initial conditions, we consider $y_1(0) = 2$ and $y_2(0) = 0$. For our computations we set the time intervals to be $[0,6]$.

For the implementation of the proposed method, we defined a trial solution satisfying the initial conditions as follows:

$$\Psi_1(t,w_1^o) = \alpha_1^{(0)} + tN_1(t,w_1^o), \Psi_2(t,w_2^o) = \alpha_2^{(0)} + tN_2(t,w_2^o), \tag{5.23}$$

where $\alpha_1^{(0)} = 2$ and $\alpha_2^{(0)} = 0$.

We carried out numerical experiments within a wide range of values of the parameter μ that controls the stiffness, namely from $\mu = 1$ to $\mu = 100000$, using a constant grid of $n = 60$ equidistant points in the interval $[0,6]$ for the proposed machine learning scheme. For the sake of presentation, in the following Tables, we show here only the results obtained for $\mu = 1,5,6,10,100,1000,10000,100000$, which are representative of the behaviour of of the methods, while in the following Figures we show only the solutions obtained for $\mu = 1,5,6,10$, where steep gradients arise; for $\mu > 10$ the solutions become flatter in the interval of interest [0, 6]. We tested the numerical performance of the proposed method against `ode45` and `ode15s`. Here, for implementing our scheme, we set $n = 60$ and tested its performance using two values for the tolerance tol, i.e. 10^{-2} and 10^{-3}. For comparison purposes, we used the same values of both the absolute and relative error tolerances for the functions `ode45` and `ode15s`. The reference solution was computed using `ode15s` with tolerances equal to 10^{-14}. The resulting approximated solutions for $\mu = 1,5,6,10$ are depicted in Figure 5.4 for the tolerance of 10^{-2} and in Figure-5.2 for 10^{-3}. The corresponding approximation errors for the various values of the parameter μ, in terms of the L_2-norm, the L_∞-norm of the error and in terms of the

MAE, are reported in Tables 5.3-5.6. Again, the errors in these tables were computed using the grid points chosen to train the proposed RPNN. In Table 5.7, we provide the number of points required by each method and in Tables 5.8- 5.9, we report the corresponding execution times (median, 5th percentile and 95th percentile) for the tolerances of 10^{-2} and 10^{-3}, respectively, including the time required to compute the reference solution, for various values of the parameter μ.

As shown, for the case when tolerances are set to 10^{-2} and for relatively small values of μ (indicatively, $\mu < 10$), the proposed machine learning scheme outperforms `ode15s` in all metrics, while its performance is comparable to the one obtained with `ode45` (see Tables 5.3-5.6). It is worthy to note that for small values of μ, where the solution exhibits steep gradients in the specific time interval (e.g. for $\mu = 5,6$), `ode15s` gives relatively large errors. Indicatively, for $\mu = 6$, the L_∞-norm of the difference between the value of y_1 (y_2) computed by `ode15s` and the corresponding component of the reference solution at the end of the interval ($t = 6$), where a steep gradient arises, is $\|y_1 - y_1^{ref}\|_{L_\infty} \simeq 1$ ($\|y_2 - y_2^{ref}\|_{L_\infty} \simeq 4.53$), while the reference values is $y_1^{ref} \simeq -0.74$ ($y_2^{ref} \simeq -8.6$). That is, `ode15s` actually fails to adequately approximate the solution. The corresponding value of L_∞-norm of the approximation errors resulting from the implementation of the proposed scheme is $\|y_1 - y_1^{ref}\|_{L_\infty} \simeq 0.13$ ($\|y_2 - y_2^{ref}\|_{L_\infty} \simeq 0.46$). On the other hand, when the value of the stiffness parameter is relatively small, `ode45` results to considerably larger approximation errors when compared to our scheme. In particular, for $\mu = 1$, the L_{inf}-norm of the difference between the value of y_1 (y_2) provided by `ode45` and the reference solution at the end of the interval is $\|y_1 - y_1^{ref}\|_{L_\infty} \simeq 0.13$ ($\|y_2 - y_2^{ref}\|_{L_\infty} \simeq 0.14$), with reference value $y_1^{ref} \simeq 1.27$ ($y_2^{ref} \simeq 2.44$). For $\mu = 1$, the corresponding norms obtained with the proposed scheme are $\|y_1 - y_1^{ref}\|_{L_\infty} \simeq 0.01$ ($\|y_2 - y_2^{ref}\|_{L_\infty} \simeq 0.01$). For larger values of the stiffness parameter (e.g. for $\mu > 100$) `ode15s` results to the best approximation accuracy; as shown, the proposed scheme achieves an adequate numerical accuracy ranging within the orders of 10^{-5} to 10^{-7}.

Interestingly enough, when the tolerances are set equal to 10^{-3}, we have a behaviour similar to the case when the tolerance is set to 10^{-2}. In particular, for solutions with steep gradients as in the case of $\mu = 6$, the proposed method provides better approximations when compared to `ode15s` and almost the same with `ode45`. For larger values of the stiffness parameter (e.g. for $\mu > 100$) `ode15s` results to the best approximation accuracy, yet, the proposed scheme achieves adequate numerical accuracy ranging between 10^{-5} and 10^{-7}.

Thus, we conclude that our RPNN-based approach performs well in terms of accuracy regardless of the level of stiffness, combining in some sense the “good" properties of the other two solvers.

Table 5.3 Van der Pol problem. Absolute errors (L_2-norm, L_∞-norm and MAE) for the component y_1 of the solutions computed with tolerances set to 1e−02. The reference solution was obtained with `ode15s` with tolerances set to 1e−14.

	L_2	L_∞	MAE	L_2	L_∞	MAE	L_2	L_∞	MAE
	$\mu = 1$			$\mu = 5$			$\mu = 6$		
RPNN	0.0234	0.0143	0.0012	0.1443	0.0891	0.0070	0.1652	0.1284	0.0043
`ode45`	0.5091	0.1746	0.0433	0.1250	0.0790	0.0043	0.1486	0.1112	0.0054
`ode15s`	0.1534	0.0565	0.0123	1.0602	0.6514	0.0521	1.5121	0.9553	0.0654
	$\mu = 10$			$\mu = 100$			$\mu = 1000$		
RPNN	0.0013	0.0006	0.0002	0.0003	6.58e-5	3.54e-5	9.09e-5	2.37e-5	9.96e-6
`ode45`	0.0017	0.0007	0.0002	0.0001	4.73-5	1.53e-5	9.94e-6	3.43e-6	8.87e-7
`ode15s`	0.0179	0.0035	0.0020	0.0001	2.69e-5	9.49e-6	6.19e-8	1.68e-8	6.35e-9
	$\mu = 10000$			$\mu = 100000$					
RPNN	1.01e-5	2.34e-6	1.11e-6	6.36e-7	1.33e-7	7.99e-8			
`ode45`	1.04e-6	3.31e-7	9.34e-8	1.07e-7	3.68e-8	9.98e-9			
`ode15s`	1.45e-10	4.11e-11	1.51e-11	2.56e-12	7.01e-13	2.60e-13			

5.3.3 Case study 3: ROBER problem

The ROBER model was developed to describe the kinetics of an autocatalytic reaction [110]. The set of the reactions reads:

$$A \xrightarrow{k_1} C + B, B + B \xrightarrow{k_2} C + B, B + C \xrightarrow{k_3} A + C, \tag{5.24}$$

where A, B, C are chemical species and k_1, k_2 and k_3 are reaction rate constants. Assuming idealized conditions and the mass action law is applied for the rate functions, we have the following system of ODEs:

$$\begin{aligned} y_1' &= -k_1 y_1 + k_3 y_2 y_3, \\ y_2' &= k_1 y_1 - k_3 y_2 y_3 - k_2 y_2^2, \\ y_3' &= k_2 y_2^2, \end{aligned} \tag{5.25}$$

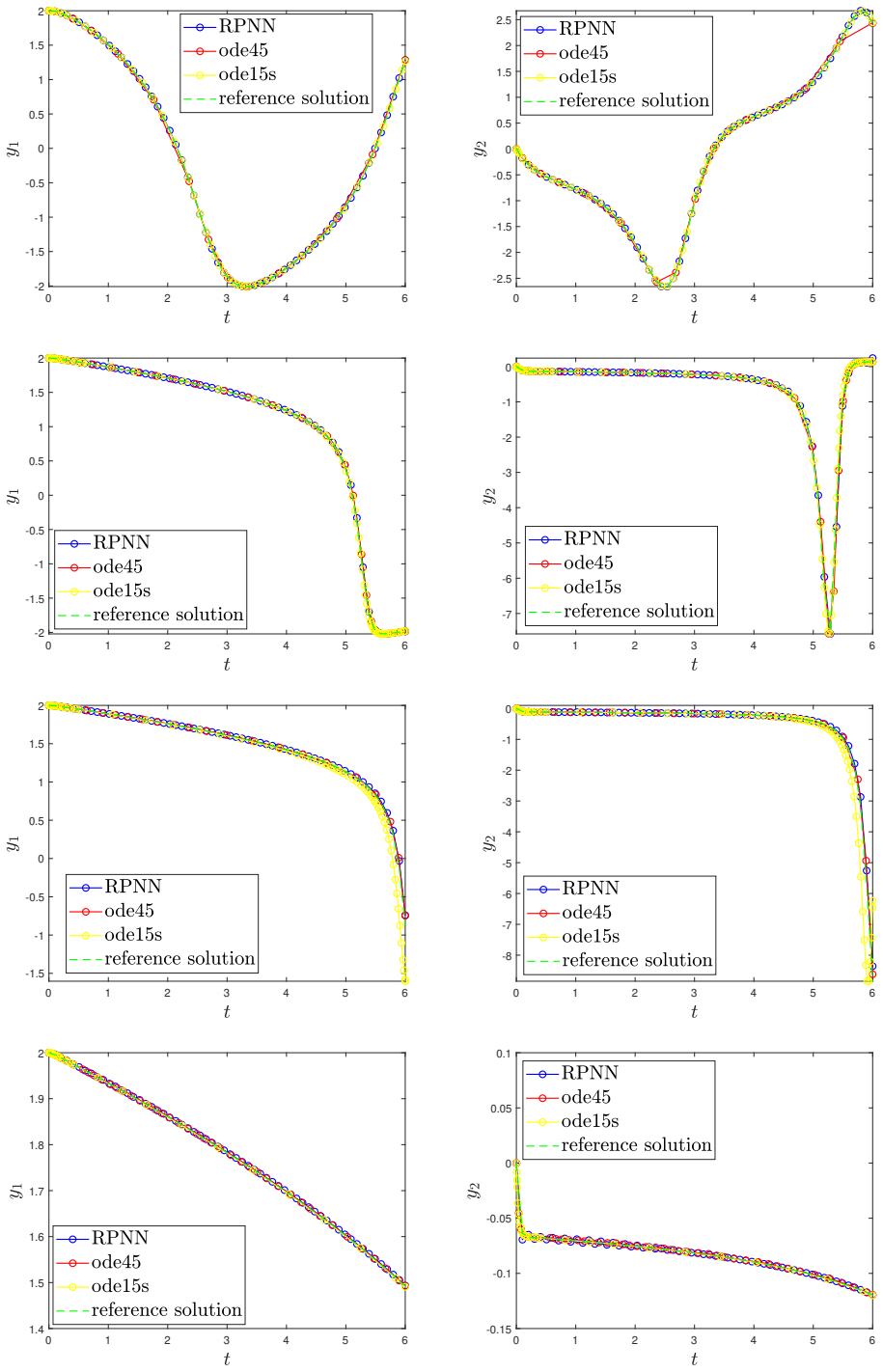


Fig. 5.2 Numerical solutions of the van der Pol problem in the interval $[0,6]$ with tolerances set to $tol = 1e-3$: (a)-(b) $\mu = 1$, (c)-(d) $\mu = 5$, (e)-(f) $\mu = 6$, and (g)-(h) $\mu = 10$. y_1 is shown on the left ((a),(c),(e) and (g)) and y_2 on the right ((b),(d),(f) and (h)). The reference solution was obtained with `ode15s` with tol= 1e−14.

Table 5.4 Van der Pol problem. Absolute errors (L_2-norm, L_∞-norm and MAE) for the component y_2 of the solutions computed with tolerances set to 1e−02. The reference solution was obtained with `ode15s` with tolerances set to 1e−14.

	L_2	L_∞	MAE	L_2	L_∞	MAE	L_2	L_∞	MAE
	$\mu = 1$			$\mu = 5$			$\mu = 6$		
RPNN	0.0432	0.0165	0.0041	0.7567	0.6023	0.0318	0.7029	0.4613	0.0233
`ode45`	0.4769	0.1993	0.0433	0.6919	0.5459	0.0281	0.5303	0.4275	0.0192
`ode15s`	0.3483	0.1321	0.0261	5.0013	3.0131	0.2172	6.6734	4.5342	0.2458
	$\mu = 10$			$\mu = 100$			$\mu = 1000$		
RPNN	0.0190	0.0123	0.0011	0.0045	0.0028	0.0003	0.0011	0.0003	9.19-5
`ode45`	0.0290	0.0104	0.0025	0.0332	0.0106	0.0033	0.0299	0.0113	0.0027
`ode15s`	0.0017	0.0006	0.0002	1.38e-6	1.18e-6	8.52e-8	8.18e-10	1.43e-10	9.54e-11
	$\mu = 10000$			$\mu = 100000$					
RPNN	3.34e-5	2.32e-5	2.19e-6	4.01e-6	2.11e-6	2.77e-7			
ode45	0.0321	0.0103	0.0029	0.0308	0.0106	0.0028			
ode15s	5.58e-14	4.43e-14	4.06e-15	7.77e-16	1.26e-16	8.82e-17			

Table 5.5 Van der Pol problem. Absolute errors (L_2-norm, L_∞-norm and MAE) for the component y_2 of the solutions computed with tolerances set to $1e-03$. The reference solution was obtained with `ode15s` with tolerances set to 1e−14.

	L_2	L_∞	MAE	L_2	L_∞	MAE	L_2	L_∞	MAE
	$\mu = 1$			$\mu = 5$			$\mu = 6$		
RPNN	0.0254	0.0097	0.0021	0.1432	0.0902	0.0071	0.1723	0.1349	0.0061
`ode45`	0.0294	0.0111	0.0024	0.1402	0.0874	0.0068	0.1672	0.1311	0.0067
`ode15s`	0.0116	0.0032	0.0012	0.0350	0.0217	0.0018	1.0857	0.7257	
	$\mu = 10$			$\mu = 100$			$\mu = 1000$		
RPNN	0.0013	0.0006	0.0001	0.0012	0.0002	0.0001	8.23e-5	2.52e-5	9.18e-6
`ode45`	0.0013	0.0004	0.0001	9.42e-5	1.75e-5	1.13e-5	2.21e-6	6.35e-7	2.30e-7
`ode15s`	0.0074	0.0013	0.0009	0.0001	2.74e-5	9.94e-6	1.20e-7	2.91e-8	1.26e-8
	$\mu = 10000$			$\mu = 100000$					
RPNN	6.94e-6	1.31e-6	8.59e-7	1.18e-6	3.89e-7	1.27e-7			
`ode45`	1.19e-7	3.90e-8	1.16e-8	1.08e-8	3.49e-9	9.95e-10			
`ode15s`	6.47e-10	1.85e-10	6.54e-11	7.56e-12	1.89e-12	8.15e-13			

where y_1, y_2 and y_3 denote the concentrations of A, B and C, respectively. In our simulations, we set the typical values of the parameters, i.e. $k_1 = 0.04$, $k_2 = 10^4$

Table 5.6 Van der Pol problem. Absolute errors (L_2-norm, L_∞-norm and MAE) for the component y_2 of the solutions computed with tolerances set to 1e−03. The reference solution was obtained with `ode15s` with tolerances set to 1e−14.

	L_2	L_∞	MAE	L_2	L_∞	MAE	L_2	L_∞	MAE
	$\mu = 1$			$\mu = 5$			$\mu = 6$		
RPNN	0.0468	0.0158	0.0041	0.7537	0.6011	0.0308	0.7295	0.4824	0.0254
`ode45`	0.0554	0.0198	0.0045	0.6954	0.5114	0.0288	0.6176	0.4947	0.0212
`ode15s`	0.0153	0.0051	0.0015	0.1711	0.1237	0.0075	4.6391	2.9251	0.1703
	$\mu = 10$			$\mu = 100$			$\mu = 1000$		
RPNN	0.0101	0.0088	0.0005	0.0050	0.0022	0.0004	0.0008	0.0003	6.78e-5
`ode45`	0.0029	0.0010	0.0003	0.0035	0.0011	0.0003	0.0036	0.0011	0.0003
`ode15s`	0.0009	0.0003	8.98e-5	5.27e-7	1.59e-7	5.37e-8	8.07e-10	1.55e-10	9.11e-11
	$\mu = 10000$			$\mu = 100000$					
RPNN	3.21e-5	2.12e-5	2.43e-6	1.21e-5	3.75e-6	1.09e-6			
`ode45`	0.0036	0.0011	0.0004	0.0032	0.0010	0.0003			
`ode15s`	2.99e-13	1.05e-13	2.62e-14	7.51e-16	1.30e-16	8.59e-17			

Table 5.7 Van der Pol problem in the interval $[0,6]$. Number of collocation points required from each method with tolerances set to 1e−02 and 1e−03. The reference solution was obtained with `ode15s` with tolerances set to 1e−14. The number of points used in the RPNN method were fixed to 60 regardless of the value of μ.

	μ	1	5	6	10	100	1000	10000	100000
$tol = 10^{-2}$	`ode45`	13	25	22	41	530	5410	54221	542336
	`ode15s`	35	52	33	20	19	20	21	23
$tol = 10^{-3}$	`ode45`	16	30	25	43	530	5411	54222	542336
	`ode15s`	54	71	65	28	24	22	23	24
	reference solution	356	455	301	146	99	87	66	49

and $k_3 = 3 \times 10^7$, with $y_1(0) = 1$, $y_2(0) = 0$ and $y_3(0) = 0$ as initial values of the concentrations, and we consider the time interval $[0, 1.8]$.

Based on the proposed methodology, we construct an (initial) trial solution that satisfies the initial conditions with $\alpha_1^{(0)} = 1$, $\alpha_2^{(0)} = 0$ and $\alpha_3^{(0)} = 0$. For the numerical solution of the ROBER system with the proposed method, we used a grid of $n = 14$ equidistant collocation points in the time interval $[0, 1.8]$. For our illustrations, we tested the numerical performance of the scheme using two values of the tolerance

Table 5.8 van der Pol problem. Median, 5th and 95th percentiles of the computational time (in seconds) for the different methods with tolerances set to 10^{-2} for various values of μ. The reference solution is computed with `ode15s` when tolerances equal to 10^{-14}.

	median	5th perc	95th perc	median	5th perc	95th perc	median	5th perc	95th perc
	$\mu = 1$			$\mu = 5$			$\mu = 6$		
RPNN	0.1578	0.1448	0.1960	0.2052	0.1806	0.2409	0.2269	0.2032	0.2854
`ode45`	0.0008	0.0007	0.0010	0.0009	0.0008	0.0013	0.0009	0.0008	0.0012
`ode15s`	0.0047	0.0042	0.0056	0.0081	0.0075	0.0097	0.0058	0.0053	0.0066
reference solution	0.0075	0.0068	0.0082	0.0139	0.0123	0.0161	0.0099	0.0087	0.0112
	$\mu = 10$			$\mu = 100$			$\mu = 1000$		
RPNN	0.1127	0.1060	0.1308	0.0744	0.0568	0.0868	0.0594	0.05264	0.1121
`ode45`	0.0012	0.0010	0.0016	0.0046	0.0040	0.0063	0.0421	0.0308	0.0719
`ode15s`	0.0029	0.0026	0.0037	0.0038	0.0032	0.0051	0.0040	0.0029	0.0075
reference solution	0.0049	0.0043	0.0068	0.0052	0.0044	0.0069	0.0057	0.0042	0.01163
	$\mu = 10000$			$\mu = 100000$					
RPNN	0.0278	0.0254	0.0381	0.0412	0.0367	0.0823			
`ode45`	0.3081	0.2975	0.3724	3.2561	3.0868	10.1607			
`ode15s`	0.0031	0.0034	0.0047	0.0053	0.0041	0.0143			
reference solution	0.0046	0.0042	0.0055	0.0051	0.0044	0.0153			

tol, i.e. 10^{-2} and 10^{-3}. For comparison purposes, we used the same values of both the absolute and relative error tolerances for the functions `ode45` and `ode15s`. The approximate solutions obtained with tolerances 10^{-2} and 10^{-3} are shown in figure 5.3. Furthermore, in tables 5.11 and 5.12 we report the corresponding numerical approximation accuracy obtained with the various methods in terms of the MAE, L2-norm and L∞-norm of the error with respect to the reference solution. In table 5.10, we show the number of points required by each method, and the execution times (median, 5$thpercentile$ and 95$thpercentile$) of all the methods, including the time required to compute the reference solution.

As it is shown, with tolerances equal to 10−2 the proposed machine learning method achieves more accurate solutions than `ode15s`, while `ode45` fails to converge (see tables 5.11 and 5.11). The number of points used by our approach is smaller than to those used by the other two methods (see table 5.10). On the other hand, the

Table 5.9 van der Pol problem. Median, 5th and 95th percentiles of the computational time (in seconds) for the different methods with tolerances set to 10^{-3} for various values of μ. The reference solution is computed with `ode15s` when tolerances equal to 10^{-14}.

	median	5th perc	95th perc	median	5th perc	95th perc	median	5th perc	95th perc
	$\mu = 1$			$\mu = 5$			$\mu = 6$		
RPNN	0.2219	0.2134	0.2386	0.2886	0.2512	0.3487	0.2744	0.2651	0.2913
`ode45`	0.0010	0.0009	0.0012	0.0011	0.0010	0.0013	0.0010	0.0010	0.0012
`ode15s`	0.0069	0.0063	0.0088	0.0100	0.0095	0.0121	0.0099	0.0093	0.0115
	$\mu = 10$			$\mu = 100$			$\mu = 1000$		
RPNN	0.1467	0.1402	0.1579	0.0752	0.0706	0.0818	0.0554	0.0521	0.0623
`ode45`	0.0011	0.0011	0.0014	0.0043	0.0040	0.0053	0.0313	0.0286	0.0374
`ode15s`	0.0034	0.0031	0.0039	0.0036	0.0033	0.0041	0.0037	0.0032	0.0045
	$\mu = 10000$			$\mu = 100000$					
RPNN	0.0590	0.0553	0.0651	0.0410	0.0378	0.0449			
`ode45`	0.2915	0.2806	0.3640	2.9382	2.8654	3.3288			
`ode15s`	0.0035	0.0034	0.0050	0.0048	0.0041	0.0057			

computing time of our method is larger, but we believe this is paid off by the fact that our method does not require the user to explicitly take into account the stiffness of the ODE problem and it provides an approximate solution in the form of a function that can be evaluated at points different from the collocation ones. Finally, we believe that the code implementing the proposed scheme can be made more efficient, but this is beyond the scope of the current thesis.

When the tolerances are set equal to 10^{-3}, the proposed method provides a solution whose error is generally comparable or slightly worse than the errors in the solutions computed by the other two methods. We also note that `ode45` is now able to compute an approximate solution to the problem. The execution time required by the RPNN method is again larger than the times of the classical methods, but the comments made for the case of the tolerances equal to 10^{-2} still apply.

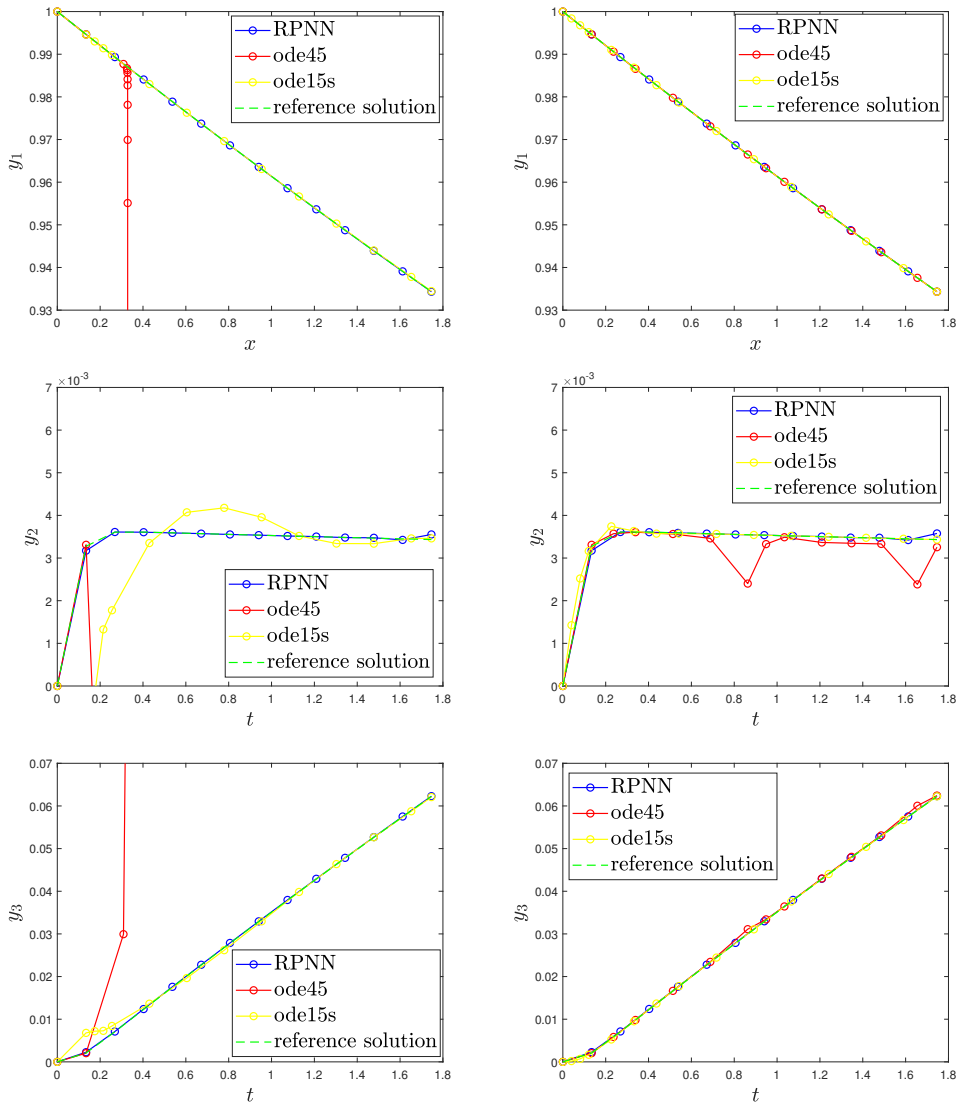


Fig. 5.3 ROBER problem. Approximate solutions computed by the different methods in the interval $[0, 1.8]$ with tolerances set to 1e−02 ((a), (c) and (e)) and 1e−03 ((b), (d) and (f)). The reference solution was obtained with `ode15s` with tolerances set to 1e−14.

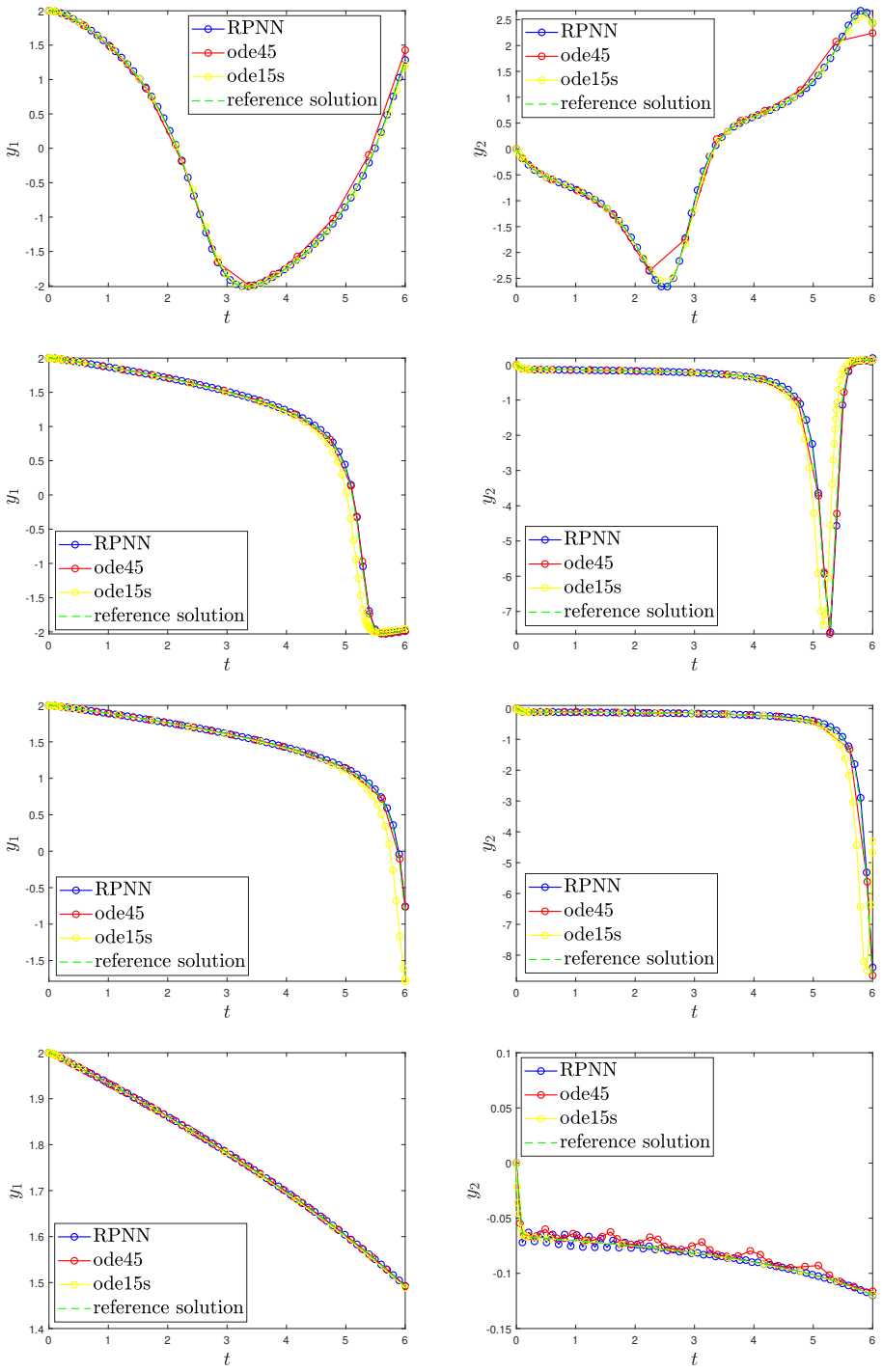


Fig. 5.4 Numerical solutions of the van der Pol problem in the interval $[0,6]$, with *tol*=1e−2: (a)-(b) $\mu = 1$, (c)-(d) $\mu = 5$, (e)-(f) $\mu = 6$, and (g)-(h) $\mu = 10$. y_1 is shown on the left ((a),(c),(e) and (g)) and y_2 on the right ((b),(d),(f) and (h)). The reference solution was obtained with `ode15s` with *tol*= 1e−14.

Table 5.10 ROBER problem. Computational times and number of points required in the interval [0, 1.8] by RPNN, `ode45` and `ode15s` with tolerances 1e−02 and 1e−03. The reference solution was computed by `ode15s` with tolerances equal to 1e−14.

	$tol = 1e{-}02$				$tol = 1e{-}03$			
	median	min	max	# pts	median	min	max	# pts
RPNN	0.0265	0.0210	0.7435	14	0.634	0.0956	2.1890	
`ode45`	-	-	-	-	4.54e−01	5.56e−01	0.0024	23
`ode15s`	0.0047	0.0034	0.0065	24	0.0033	0.0024	0.0056	22
reference	0.0053	0.0038	0.0082	191	0.0053	0.0038	0.0082	34

Table 5.11 ROBER problem. Absolute errors (L_2-norm, L_∞-norm and MAE) for the solutions computed with tolerance set to $1e-02$. The reference solution was obtained with `ode15s` with tolerances set to 1e−14.

	y_1			y_2			y_3		
	L_2	L_∞	MAE	L_2	L_∞	MAE	L_2	L_∞	MAE
RPNN	3.45e-5	3.076e-5	4.78e-6	0.0001	0.0001	1.93e-5	0.0004	0.0002	8.99e-5
`ode45`	—	—	—	—	—	—	—	—	—
`ode15s`	0.0003	0.0001	5.14e-5	0.0051	0.0024	0.0003	0.0042	0.0023	0.0003

Table 5.12 ROBER problem. Absolute errors (L_2-norm, L_∞-norm and MAE) for the solutions computed with tolerance set to $1e-03$. The reference solution was obtained with `ode15s` with tolerances set to 1e−14.

	y_1			y_2			y_3		
	L_2	L_∞	MAE	L_2	L_∞	MAE	L_2	L_∞	MAE
RPNN	9.12e-6	7.02e-6	2.45e-6	0.002	0.002	0.0005	0.0006	0.0004	0.0003
`ode45`	7.34e-7	4.50e-7	2.69e-7	9.98e-5	9.84e-5	1.76e-5	9.99e-5	9.91e-5	1.78e-5
`ode15s`	6.15e-5	4.03e-5	1.32e-5	0.0002	0.0001	3.33e-5	0.0002	0.0001	4.90e-5

5.4 Discussion

We propose a machine learning algorithm for the solution of ODEs with a focus on stiff ODEs, which combines the speed and the generalization advantages of RPNNs. In this algorithm, only the parameters from the hidden to the output layer have to be determined, and every hidden unit is “responsible” of learning the behaviour of the underlying physical model around a center point.

Theoretical results stemming from the Johnson and Lindenstrauss Theorem and universal approximation theorems that have been proved for ELMs guarantee the approximation capabilities of our RPNN, despite the much simpler way of obtaining the weights of the network.

The results show that the proposed machine learning approach is able to provide reasonably accurate numerical approximations to the solutions of four benchmark stiff ODE problems. Our algorithm outperforms `ode15s` in terms of numerical accuracy in cases where steep gradients arise in the solution, while it is more robust than `ode45`, which in turn is generally less accurate and needs more points to converge, or even fails for highly-stiff problems.

The computational times are in general larger than those required by the `ode45` and `ode15s` solvers, but comparable for all practical purposes. However, our method results in smaller computational times than `ode45` and `ode15s` when the solution has to be computed in "dense" sets of points, because the method provides an analytical expression for the approximation of the solution upon training with a finite set of collocation points. Furthermore, in our opinion the larger computational times are also due to the fact that our home-made code is not optimized (its optimization is out of the scope of the current thesis).

Chapter 6

Constructing coarse-scale bifurcation diagrams from spatio-temporal observations of microscopic simulations. A machine learning approach.

6.1 Methodology

The pipeline of our computational framework for constructing the bifurcation diagrams of the hidden coarse-scale PDEs from data coming from detailed microscopic simulations consists of three tasks: (a) identify the coarse-scale variables from fine-scale time series, (b) learn the coarse-scale PDEs using machine learning and, (c) wrap around the machine learning model an appropriate numerical bifurcation method to construct the corresponding coarse-grained bifurcation diagram [95].

We assume that the emergent dynamics of the complex system under study on a domain $\Omega \times [t_0, t_{end}] \subseteq \mathbb{R}^d \times \mathbb{R}$ can be modelled by a system, say of m PDEs of the

form:

$$\frac{\partial u^{(i)}(x,t)}{\partial t} \equiv u_t^{(i)} = F^{(i)}(t,x,u(x,t),\mathscr{D}u(x,t),\mathscr{D}^2u(x,t),\ldots,\mathscr{D}^{\nu}u(x,t),\varepsilon), \\ (x,t)\in\Omega\times[t_0,t_{end}], \qquad i=1,2,\ldots,m \tag{6.1}$$

where $u(x)=[u^{(1)}(x),\ldots,u^{(m)}(x)]$, $F^{(i)}$, $i=1,2,\ldots m$ is a non-linear operator, $\mathscr{D}^{\nu}u(x)$ is the generic multi-index ν-th order derivative, i.e., :

$$\mathscr{D}^{\nu}u(x) := \left\{ \frac{\partial^{|\nu|}u(x)}{\partial x_1^{\nu_1}\cdots\partial x_d^{\nu_d}} \middle| |\nu| = \nu_1+\nu_2+\cdots+\nu_d,\ \nu_1,\ldots,\nu_d\geq 0 \right\},$$

and ε denotes the (bifurcation) parameters of the system.
The boundary conditions read:

$$B_l^{(i)}(u^{(i)}(x,t)) = h_l^{(i)}(x,t) \qquad x\in\partial\Omega_l, \tag{6.2}$$

where $\{\partial\Omega_l\}$ denotes an l partition of the boundary of Ω, and initial conditions

$$u^{(i)}(t_0,x) = u_0^{(i)}, \quad x\in\Omega \tag{6.3}$$

The right-hand-side of the i-th PDE depends on say $\gamma^{(i)}$ number of variables and bifurcation parameters from the set of variables $\{x,u(x),\mathscr{D}u(x),\mathscr{D}^2u(x),\ldots,\mathscr{D}^{\nu}u(x),\varepsilon\}$. Let us denote this set as $\mathscr{S}^{(i)}$, with cardinality $|\mathscr{S}^{(i)}|=\gamma(i)$. Hence, at each spatial point $x_q, q=1,2,\ldots,M$ and time instant $t_s, s=1,2,\ldots,N$ the set of features for the i-th PDE can be described by a vector $z_q(t_s)\in\mathbb{R}^{\gamma(i)}$.
Here, we assume that such macroscopic equations in principle exist but there are not available in a closed-form.

Instead, we assume that we have detailed observations from microscopic simulations from which we can compute the time and spatial derivatives of all the observables in N points in time and M points in space using e.g. finite differences. Thus, we aim to (a) identify the intrinsic dimension of the manifold on which the coarse-grained dynamics evolve, i.e. for each PDE identify $\gamma(i)$, and the coordinates that define the low-dimensional manifold, i.e. the sets $\mathscr{S}^{(i)}$, and based on them (b) identify the right-hand-side (RHS) of the the effective PDEs using machine learning.

To demononstrate the proposed approach, we have chosen to use fine-scale data from LBM simulations of the coupled FitzHugh-Nagumo PDEs of activation and inhibition dynamics. Using the LBM simulator, we produced data in time and space from different initial conditions and values of the bifurcation parameter. For the identification of the right coarse-scale variables that define the low-dimensional manifold on which the emergent dynamics evolve, we performed feature selection using parsimonious Diffusion Maps [180, 181]. Then, we trained the machine learning schemes to learn the right-hand-side of the coarse-grained PDEs on the low-dimensional manifold. Based on the constructed model, we performed numerical bifurcation analysis, employing the pseudo-arc-length continuation method. The performance of the data-driven scheme for constructing the coarse-grained bifurcation diagram was validated against the one computed with the PDEs using finite differences. A schematic overview of the proposed framework for the the case of two effective PDEs (as in the paradigm of FHN dynamics) is shown in Figure (6.1).

In what follows, we describe the method that we used for feature selection which is based on Diffusion Maps algorithm. After that, we describe the macroscale model of the FHN equations and the microsimulator LBM model which we used for simulating fine scale observations.

6.1.1 Feature selection

Here, by identifying the coarse-scale spatio-temporal behaviour of a system of PDEs, we mean learning their right-hand sides. Hence, we first have to deal with the task of discovering the few coarse-grained spatial derivatives from the high-dimensional input data space. There exist various methods for feature selection such as LASSO ([186, 187]), Random Forests ([188, 189]). In our framework, we used a technique that extracts the dominant features based on manifold parametrization through output-informed Diffusion Maps ([180]). The core assumption of this method is that if a given dataset in a high-dimensional, then using manifold learning algorithms (here Diffusion Maps), we can parametrize an ambient Euclidean space that lies on a lower-dimensional manifold.
For this purpose, given a set of $\phi_1, \phi_2, \ldots, \phi_{k-1} \in \mathbb{R}^N$ Diffusion Maps eigenvectors, for each element $i = 1, 2 \ldots, N$ of ϕ_k, we use a local linear regression model:

$$\phi_{k,i} \approx \alpha_{k,i} + \beta_{k,i}^T \Phi_{k-1,i}, \quad i = 1, 2, \ldots, N \tag{6.4}$$

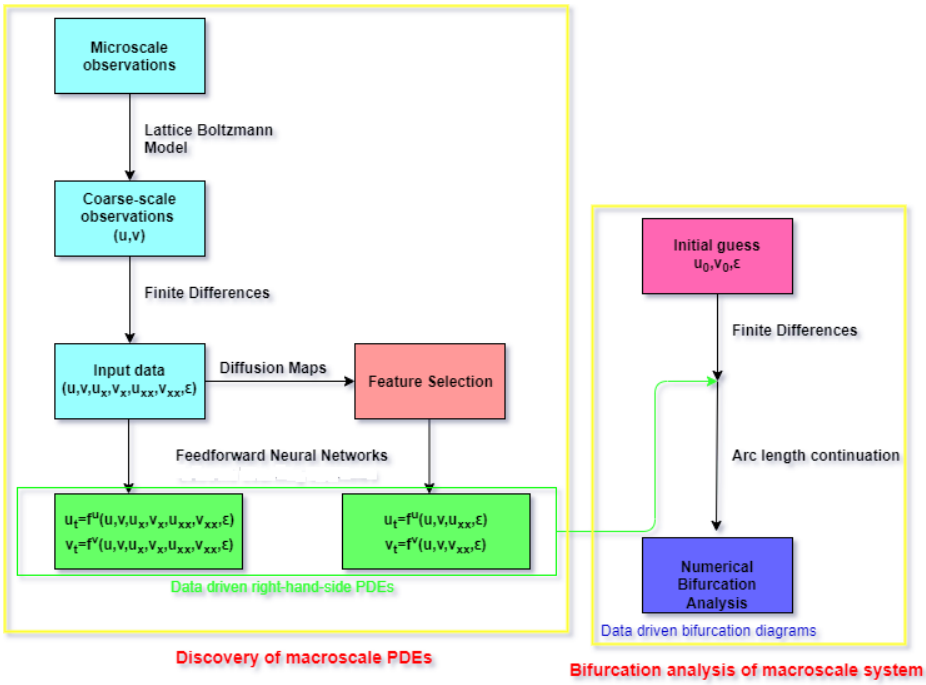


Fig. 6.1 Schematic of the three-stage workflow for constructing coarse-grained bifurcation diagrams from fine scale observations using the paradigm of two PDEs: 1) Identify the parsimonious coarse-grained observables using Diffusion Maps from microscopic simulations (here Lattice-Boltzmann) and compute their spatial derivatives using finite differences, 2) "learn" the right hand side PDEs using machine learning algorithms (here FNNs and RPNN) and 3) employ the tools of numerical bifurcation analysis (here the pseudo arc-length continuation method to construct the coarse-grained bifurcation diagrams.

to investigate if ϕ_k is an dependent eigendirection; $\Phi_{k-1,i} = [\phi_{1,i}, \phi_{2,i}, \ldots, \phi^T_{k-1,i}]$, $\alpha_{k,i} \in \mathbb{R}$ and $\beta_{k,i} \in \mathbb{R}^{k-1}$. The values of parameters $\alpha_{k,i}$ and $\beta_{k,i}$ are found solving an optimization problem of the form:

$$\hat{\alpha}_{k,i}, \hat{\beta}_{k,i} = \underset{\alpha,\beta}{\operatorname{argmin}} \sum_{j \neq i} K(\Phi_{k-1,i}, \Phi_{k-1,j})(\phi_{k,j} - (\alpha + \beta^T \Phi_{k-1,j}))^2 \tag{6.5}$$

where K is a kernel weighted function, usually the Gaussian kernel:

$$K(\Phi_{k-1,i}, \Phi_{k-1,j}) = \exp\left(-\frac{||\Phi_{k-1,i} - \Phi_{k-1,j}||}{\sigma^2}\right), \tag{6.6}$$

where σ is the shape parameter. The final normalized leave-one-out cross-validation (LOOCV) error for this local linear fit is defined as:

$$r_k = \sqrt{\frac{\sum_{i=1}^{N} (\phi_{k,i} - (\hat{\alpha}_{k,i} + \hat{\beta}^T_{k,i} \Phi_{k-1,i})^2}{\sum_{i=1}^{\mu} (\phi_{k,i})^2}}. \tag{6.7}$$

For small values of r_k, ϕ_k is considered to be dependent of the other eigenvectors and hence as a harmonic or repeated eigendirection, while large values of r_k, suggest that ϕ_k can serve as a new independent eigendirection.

In our application, we provide as inputs to the diffusion algorithm the combined input-output domain (the observables z_i and their spatial and time derivatives). In principle, any of the subsets that is capable to parametrize the discovered embedding coordinates that were chosen after the above described methodology, can be considered as a new possible input data domain that can be used for learning the right-hand-side PDE. We find the subsets that minimally parametrize the intrinsic embeddings by quantifying it with a total regression loss (L_T) based on a mean squared error:

$$L_T = (\sum_{k=1}^{\mu} L^2_{\phi_k})^{\frac{1}{2}}. \tag{6.8}$$

Here, as L_{ϕ_j} we define the Gaussian process regression loss ([190]) for representing the intrinsic coordinate ϕ_j when using s out of n selected input features:

$$L_{\phi_k} = \frac{1}{N} \sum_{i=1}^{N} (\phi_{k,i} - g(\cdot))^2, \tag{6.9}$$

where $g(\cdot)$ is the output of the Gaussian process regressor with inputs the values of the features in the ambient space and target values the eigenvectors ϕ_k. Note that for this procedure we don't include the bifurcation parameter to the dataset. We have chosen to employ the above method separately for every subset of the same value of the bifurcation parameter and finally to select the subset of features with the minimum sum of total regression loses across all the embedding spaces.

6.1.2 The macroscale model: The Fitzhugh-Nagumo partial differential equations

For demonstrating the performance of the proposed scheme, we selected the celebrated, well studied FitzHugh-Nagumo (FHN) model first introduced in [191] to simplify the Hodgkin-Huxley model into a two-dimensional system of ODEs to describe the dynamics of the voltage across a nerve cell. In particular, we consider the FHN equations which adds a spatial diffusion term to describe the propagation of an action potential as a traveling wave. The bifurcation diagram of the one-dimensional two Partial Differential Equations (PDEs) is known to have a turning point and two supercritical Andronov-Hopf bifurcation points. In what follows, we provide the model along with the initial and boundary conditions and then we describe the microscopic simulator constructed based on the Lattice-Boltman modeling approach.

The system consist in two coupled nonlinear reaction-diffusion PDEs, describing the evolution of an activator $u : [x_0, x_{end}] \times [t_0, t_{end}] \to \mathbb{R}$ and an inhibitor $v : [x_0, x_{end}] \times [t_0, t_{end}] \to \mathbb{R}$:

$$\begin{aligned} \frac{\partial u}{\partial t} &= D^u \frac{\partial^2 u}{\partial x^2} + u - u^3 - v, \\ \frac{\partial v}{\partial t} &= D^v \frac{\partial^2 v}{\partial x^2} + \varepsilon(u - \alpha_1 v - \alpha_0), \end{aligned} \tag{6.10}$$

where α_0 and α_1 are model parameters, ε is the kinetic bifurcation parameter and D^u, D^v are the diffusion coefficients for u and v respectively. For our simulations, we set $\alpha_1 = 2, \quad \alpha_0 = -0.03, \quad D^u = 1, \quad D^v = 4$ (as described in [192]) and vary the bifurcation parameter ε in the interval $[0.005, 0.955]$ concentrating the points at the boundary by taking a grid of Chebychev-Gauss-Lobatto nodes. We use a uniform discretization of the spatial domain $[x_0, x_{end}] = [0, 20]$ with a step $\Delta x = 0.2$ and of the time domain on $[t_0, t_{end}] = [0, 450]$ with a time step $\Delta t = 0.01$. To solve the equation

we consider Homogeneous Neumann Boundary conditions at both boundaries:

$$\begin{aligned} &\frac{du}{dx}(0,t) = 0, \quad \frac{dv}{dx}(0,t) = 0. \\ &\frac{du}{dx}(20,t) = 0, \quad \frac{dv}{dx}(20,t) = 0. \end{aligned} \tag{6.11}$$

and various initial conditions $u_0(x) = u(x,0)$ and $v_0(x) = v(x,0)$, that, in order to explore different behaviours, we randomly select varying the following parameters of an hyperbolic tangent sigmoid:

$$\begin{aligned} &u_0(x) = w \tanh\big(\alpha(x-c)\big) + \beta \\ &v_0(x) = 0.12 \cdot u_0(x). \\ &w \sim \mathscr{U}(0.8, 1.2), \qquad \alpha \sim \mathscr{U}(0.5, 1) \\ &c \sim \mathscr{U}(2, 18), \qquad \beta \sim \mathscr{U}(-0.4, 0), \end{aligned} \tag{6.12}$$

where $\mathscr{U}(a,b)$ is used to denote the uniform distribution in the interval $[a,b]$.

6.1.3 The microscale simulator: the Lattice Boltzman model for the Fithugh-Nagumo PDEs

Lattice-Boltzman modeling ([5, 193, 194]) can be used as a mesoscopic numerical simulation for identifying spatiotemporal dynamics of finite-difference-type discretizations of the Boltzman-BGK equations ([195]). In this paper, the lattice Boltzmann model is our fine-scale "microscopic simulator", and its results are considered to be "the ground truth" from which the coarse-scale PDE and the coarse-scale bifurcation diagram will be learned.

The microscopic proprieties of atoms and molecules determine the macroscopic evolution of a physical system, but it is generally difficult to describe a system considering equations for the dynamic of each particle. The statistical description of the system at a mesoscopic level use the concept of distribution function $f(\cdot)$:

$$\begin{aligned} f_i^u(x_{j+i}, t_{k+1}) &= f_i^u(x_j, t_k) + \Omega_i^u(x_j, t_k) + R_i^u(x_j, t_k) \\ f_i^v(x_{j+i}, t_{k+1}) &= f_i^v(x_j, t_k) + \Omega_i^v(x_j, t_k) + R_i^v(x_j, t_k), \end{aligned} \tag{6.13}$$

where Ω_i represent the collision term defined by Bhatnagar-Groos-Krook (BGK) [196]:

$$\begin{aligned}\Omega_i^u(x_j,t+k) &= -\omega^u(f_i^u(x_j,t_k) - f_i^{u,eq}(x_j,t_k)) \\ \Omega_i^v(x_j,t+k) &= -\omega^v(f_i^v(x_j,t_k) - f_i^{v,eq}(x_j,t_k)).\end{aligned} \tag{6.14}$$

ω is a BGK relaxation coefficient defined as [197]:

$$\begin{aligned}\omega^u &= \frac{2}{1+3D^u\frac{\Delta t}{\Delta x^2}} \\ \omega^v &= \frac{2}{1+3D^v\frac{\Delta t}{\Delta x^2}}.\end{aligned} \tag{6.15}$$

We use the D1Q3 (i.e. one dimension lattice where particles can stream to the right (f_1), to the left (f_{-1}) or staying still (f_0)). On the zeroth moment of the overall distribution function, one can finally compute the coarse-scale observables [5]:

$$\begin{aligned}u(x_j,t_k) &= \sum_{i=-1}^{1} f_i^u(x_j,t_k) \\ v(x_j,t_k) &= \sum_{i=-1}^{1} f_i^v(x_j,t_k).\end{aligned} \tag{6.16}$$

The coexistence of multiple distributions makes it necessary to introduce weights ω_i for the connections in the lattice and based on spatially uniform Local diffusion equilibrium, for which f_i^{eq} must be homogeneous in all velllocity directions, the weights are chosen $w_0 = 4/6, w_1 = w_{-1} = 1/6$. Thus,:

$$\begin{aligned}f_i^{u,eq}(x_j,t_k) &= w_i u(x_j,t_k) \\ f_i^{v,eq}(x_j,t_k) &= w_i v(x_j,t_k), \quad i=-1,0,1.\end{aligned} \tag{6.17}$$

Finally, the reaction terms R_i in equation 6.13 are computed by:

$$\begin{aligned}R_i^u(x_j,t_k) &= w_i\Delta t(u(x_j,t_k) - u(x_j,t_k)^3 - v(x_j,t_k)) \\ R_i^v(x_j,t_k) &= w_i\Delta t\varepsilon(u(x_j,t_k) - \alpha_1 v(x_j,t_k)^3 - \alpha_0).\end{aligned} \tag{6.18}$$

We set the model parameters equals to the macro scale FHN simulator and again we vary the bifurcation parameter ε in the interval $[0.005, 0.955]$. The spatial and time domains also remains the same as described in the previous section and we apply homogeneous Neumann boundary conditions at both boundaries and simulate the behaviours of u and v from FHN model using the mesoscale LBM simulator.

6.1.4 Algorithm

Summarizing, the proposed three-tier algorithm for constructing numerically bifurcation diagram from data is provided as pseudo code in algorithm 2 and algorithm 3. In algorithm 2 the first two steps of identifying the effective coarse scale observables and learning the dynamics are shown. In algorithm 3, we describe the third step of applying the pseudo Arc-length method to the "black-box" machine learning models.

6.2 Numerical bifurcation analysis of the FHN PDEs

For comparison purposes, we first construct the bifurcation diagram of the FHN PDEs using central Finite Differences. The discretization of the one-dimensional PDEs in M points with second-order central finite differences in the unit interval $0 \leq x \leq 200$ leads to the following system of $2*(M-2)$ algebraic equations $\forall x_j = (j-1)h, j = 2, \ldots M-1, h = \frac{1}{M-1}$:

$$F_j^u(u,v) = \frac{D^u}{h^2}(u_{j+1} - 2u_j + u_{j-1}) + u_j - u_j^3 - v_j = 0$$
$$F_j^v(u,v) = \frac{D^v}{h^2}(v_{j+1} - 2v_j + v_{j-1}) + \varepsilon(u_j - \alpha_1 v_j - \alpha_0) = 0.$$

At the boundaries, we imposed homogeneous von Neumann boundary conditions. The above $2*(M-2)$ nonlinear algebraic equations are solved iteratively using Newton's method. The non-null elements of the Jacobian matrix are given by:

$$\frac{\partial F_j^u}{\partial u_{j-1}} = \frac{D^u}{h^2}; \frac{\partial F_j^u}{\partial u_j} = -D^u\frac{2}{h^2} - 3u_j^2; \frac{\partial F_j^u}{\partial u_{j+1}} = \frac{D^u}{h^2}; \frac{\partial F_j^u}{\partial v_j} = -1$$
$$\frac{\partial F_j^v}{\partial v_{j-1}} = \frac{D^v}{h^2}; \frac{\partial F_j^v}{\partial v_j} = -D^v\frac{2}{h^2} - \varepsilon\alpha_1 v_j; \frac{\partial F_j^v}{\partial v_{j+1}} = \frac{D^u}{h^2}; \frac{\partial F_j^v}{\partial u_j} = \varepsilon.$$

Algorithm 2 Identify the coarse scale observables from fine scale observations and learn the coarse dynamics for the FHN PDEs.

Require: Grid N_ε of Chebysev-Gauss-Lobatto points in [0.005,0.955] for ε ▷ *set grid for bifurcation parameter* ε

Require: $x = [x_0 : \Delta x : x_{end}]$ and $t = [t0 : \Delta t : t_{end}]$ be the space and time grid with M and N points respectively

$inits \leftarrow N_i$ ▷ *set # of initial conditions*

1. Identify the coarse scale observables.

$L_t^u \leftarrow 0, L_t^v \leftarrow 0$

for $\varepsilon = 1, \ldots, N_\varepsilon$ **do**

  for $i = 1, \ldots, N_i$ **do**

    Select $w \sim \mathscr{U}(0.8, 1.2)$, $c \sim \mathscr{U}(2, 18)$,
    $\alpha \sim \mathscr{U}(0.5, 1)$, $\beta \sim \mathscr{U}(-0.4, 0)$.

    $u_i(x,0,\varepsilon) \leftarrow wtanh(\alpha(x-c)) + \beta$

    $v_i(x,0,\varepsilon) \leftarrow 0.12 \cdot u_i(x,0,\varepsilon)$ ▷ *see equation 6.12*

    for $s = 2 : N$ **do**

      $f_j^{u,eq}(x,t(s),\varepsilon) \leftarrow w_j u(x,t(s),\varepsilon),\ j = -1,0,1$

      $f_j^{v,eq}(x,t(s),\varepsilon) \leftarrow w_j v(x,t(s),\varepsilon),\ j = -1,0,1$ ▷ *see equation 6.17*

      $u_i(x,t(s),\varepsilon) \leftarrow \sum_{j=-1}^{1} f_j^{u,eq}(x,t(s),\varepsilon)$

      $v_i(x,t(s),\varepsilon) \leftarrow \sum_{j=-1}^{1} f_j^{v,eq}(x,t(s),\varepsilon)$ ▷ *see equation 6.16*

      Compute $u_{t,i}, v_{t,i}, u_{x,i}, v_{x,i}, u_{xx,i}, v_{xx,i}$ ▷ *using central finite differences.*

  Compute the first μ Diffusion Maps (DM) eigenvectors:

  $[\phi_1^u, \ldots, \phi_\mu^u] \leftarrow \mathrm{DM}(u, v, u_t, u_x, v_x, u_{xx}, v_{xx})$

  $[\phi_1^v, \ldots, \phi_\mu^v] \leftarrow \mathrm{DM}(u, v, v_t, u_x, v_x, u_{xx}, v_{xx})$ ▷ *see equation 4.14*

  $z \leftarrow [u, v, u_x.v_x.u_{xx}, v_{xx}]$

  for every $q \subset z$ **do**

    for $k = 1, \ldots, \mu$ **do**

      $\hat{\phi}_k^u \leftarrow GP(q)$

      $\hat{\phi}_k^v \leftarrow GP(q)$ ▷ *GP: Gaussian process regressor*

      $L_{\phi_k^u} \leftarrow \frac{1}{N}\sum_{i=1}^{N}(\phi_{k,i}^u - \hat{\phi}_{k,i}^u)^2$

      $L_{\phi_k^v} \leftarrow \frac{1}{N}\sum_{i=1}^{N}(\phi_{k,i}^v - \hat{\phi}_{k,i}^v)^2$ ▷ *see equation 6.9*

    $L_t^u(q) \leftarrow L_t^u(q) + (\sum_{j=1}^{\mu} L_{\phi_j^u}^2)^{\frac{1}{2}}$

    $L_t^v(q) \leftarrow L_t^v(q) + (\sum_{j=1}^{\mu} L_{\phi_j^v}^2)^{\frac{1}{2}}$ ▷ *see equations 6.8 and 6.9*

$z_u \leftarrow \{q^* : L_t^u(q^*) = min(L_t^u)\}$

$z_v \leftarrow \{q^* : L_t^u(q^*) = min(L_t^v)\}$ ▷ *extract the effective features*

2. Learn the coarse scale PDEs

Train two FNNs without feature selection:

$\hat{F}^u \equiv \hat{u}_t \leftarrow \mathrm{FFN}(z, \varepsilon)$

$\hat{F}^v \equiv \hat{v}_t \leftarrow \mathrm{FFN}(z, \varepsilon)$ ▷ *without feature selection*

Train two FNNs with feature selection:

$\hat{F}_r^u \equiv \hat{u}_{t,r} \leftarrow \mathrm{FFN}(z_u, \varepsilon)$

$\hat{F}_r^v \equiv \hat{v}_{t,r} \leftarrow \mathrm{FNN}(z_v, \varepsilon)$ ▷ *with feature selection*

Algorithm 3 Construct bifurcation diagrams with pseudo Arc-length continuation method.

3. Wrap around the macine learning model the pseudo Arc-Length method.

$\varepsilon_0 \leftarrow 0.005, \varepsilon_{-2} \leftarrow \varepsilon_0$, $\varepsilon_{-1} \leftarrow \varepsilon_{-2} + 5e-04$

$u_0 \leftarrow -tanh(0.5(x-9.5)), v_0 = 0.12u_0$ ▷ *set initial guesses*

$[u_{-2}; v_{-2}] \leftarrow \mathrm{NR}(u_0, v_0, \varepsilon_{-2})$

$[u_{-1}; v_{-1}] \leftarrow \mathrm{NR}(u_0, v_0, \varepsilon_{-1})$ ▷ *approximate the solutions for the first step with Newton Raphson*

$\zeta_{-2} \leftarrow [u_{-2}; v_{-2}; \varepsilon_{-2}]$, $\zeta_{-1} \leftarrow [u_{-1}; v_{-1}; \varepsilon_{-1}]$

$maxiter \leftarrow 6$ ▷ *set max # iters for Newton method*

$\delta = 1e-06$ ▷ *peturbation parameter*

$tol = 1e-06$

repeat

  $\Delta s = \sqrt{sum((\zeta_{-1}(1:2) - \zeta_{-2}(1:2))^2) + (\zeta_{-1}(3) - \zeta_{-2}(3))^2}$

  $\zeta \leftarrow 2\zeta_{-1} - \zeta_{-2}$

  $N \leftarrow \frac{\zeta_{-1}(1:2) - \zeta_{-2}(1:2)}{\Delta s} + \frac{\zeta_{-1}(3) - \zeta_{-2}(3)}{\Delta s}$

  $iter \leftarrow 0$

  repeat

    for $j = 1, \cdots, M$ **do**

      $\frac{\partial \hat{F}^u}{\partial u_j} \leftarrow \frac{\hat{F}^u(u_j, v_j, \varepsilon) - \hat{F}^u(u_j+\delta, v_j, \varepsilon)}{2\delta}$; $\frac{\partial \hat{F}^u}{\partial v_j} \leftarrow \frac{\hat{F}^u(u_j, v_j, \varepsilon) - \hat{F}^u(u_j, v_j+\delta, \varepsilon)}{2\delta}$

      $\frac{\partial \hat{F}^v}{\partial u_j} \leftarrow \frac{\hat{F}^v(u_j, v_j, \varepsilon) - \hat{F}^v(u_j+\delta, v_j, \varepsilon)}{2\delta}$; $\frac{\partial \hat{F}^v}{\partial v_j} \leftarrow \frac{\hat{F}^v(u_j, v_j, \varepsilon) - \hat{F}^v(u_j, v_j+\delta, \varepsilon)}{2\delta}$

      $\frac{\partial \hat{F}^u}{\partial \varepsilon} \leftarrow \frac{\hat{F}^u(u_j, v_j, \varepsilon) - \hat{F}^u(u_j, v_j, \varepsilon+\delta)}{2\delta}$; $\frac{\partial \hat{F}^v}{\partial \varepsilon} \leftarrow \frac{\hat{F}^v(u_j, v_j, \varepsilon) - \hat{F}^v(u_j, v_j, \varepsilon+\delta)}{2\delta}$

      $\frac{\partial N}{\partial u_j}$; $\frac{\partial N}{\partial v_j}$; $\frac{\partial N}{\partial \varepsilon}$

    Compute the Jacobian $\nabla F \leftarrow \begin{bmatrix} \nabla_u \hat{F}^u & \nabla_v \hat{F}^u & \nabla_\varepsilon \hat{F}^u \\ \nabla_u \hat{F}^v & \nabla_v \hat{F}^v & \nabla_\varepsilon \hat{F}^v \\ \nabla_u N & \nabla_v N & \nabla_\varepsilon N \end{bmatrix}$ ▷ *see eq 6.19*

    $\Delta\zeta \leftarrow (\nabla F)^\dagger F$

    $\zeta \leftarrow \zeta - \Delta\zeta$ ▷ *update solutions*

    $err \leftarrow ||F||_2$ ▷ *compute error*

    $iter \leftarrow iter + 1$

  until $(err \leq tol)$ or $(iter \geq maxiter)$

  $\zeta_{-2} \leftarrow \zeta_{-1}, \zeta_{-1} \leftarrow \zeta$

  $\tilde{u} \leftarrow \zeta(1)$, $\tilde{v} \leftarrow \zeta(2)$, $\tilde{\varepsilon} \leftarrow \zeta(3)$

until $\tilde{\varepsilon} < \varepsilon_0$

To trace the solution branch along the critical points, we used the “pseudo" arc-Length-continuation method ([198–200]). This involves the parametrization of $u(x)$, $v(x)$ and $\varepsilon(x)$ by the arc-length s on the solution branch. The solution is sought in terms of $\tilde{u}(x,s)$, $\tilde{v}(x,s)$ and $\tilde{\varepsilon}(s)$ in an iterative manner, by solving until convergence the following augmented system:

$$\begin{bmatrix} \nabla_u F^u & \nabla_v F^u & \nabla_\varepsilon F^u \\ \nabla_u F^v & \nabla_v F^v & \nabla_\varepsilon F^v \\ \nabla_u N & \nabla_v N & \nabla_\varepsilon N \end{bmatrix} \begin{bmatrix} du^{(n)}(x,s) \\ dv^{(n)}(x,s) \\ d\varepsilon^{(n)}(s) \end{bmatrix} = - \begin{bmatrix} F^u(u^{(n)}(x,s), v^{(n)}(x,s), \varepsilon^{(n)}(s)) \\ F^v(u^{(n)}(x,s), v^{(n)}(x,s), \varepsilon^{(n)}(s)) \\ N(u^{(n)}(x,s), v^{(n)}(x,s), \varepsilon^{(n)}(s)) \end{bmatrix}, \tag{6.19}$$

where

$$\nabla_\varepsilon F^u = \begin{bmatrix} \frac{\partial F_1^u}{\partial \varepsilon} & \frac{\partial F_2^u}{\partial \varepsilon} & \dots & \frac{F_N^u}{\partial \varepsilon} \end{bmatrix}^T, \nabla_\varepsilon F^v = \begin{bmatrix} \frac{\partial F_1^v}{\partial \varepsilon} & \frac{\partial F_2^v}{\partial \varepsilon} & \dots & \frac{F_M^v}{\partial \varepsilon} \end{bmatrix}^T,$$

and

$$\begin{aligned} N(u^{(n)}(x,s), v^{(n)}(x,s), \varepsilon^{(n)}(s)) = \\ (u^{(n)}(x,s) - \tilde{u}(x,s)_{-2})^T \cdot \frac{(\tilde{u}(x)_{-2} - \tilde{u}(x)_{-1})}{ds} + \\ (v^{(n)}(x,s) - \tilde{v}(x,s)_{-2})^T \cdot \frac{(\tilde{v}(x)_{-2} - \tilde{v}(x)_{-1})}{ds} + \\ (\varepsilon^{(n)}(s) - \tilde{\varepsilon}_{-2}) \cdot \frac{(\tilde{\varepsilon}_{-2} - \tilde{\varepsilon}_{-1})}{ds} - ds, \end{aligned}$$

where $(\tilde{u}(x)_{-2}$,$\tilde{v}(x)_{-2})$ and $(\tilde{u}(x)_{-1}$,$\tilde{v}(x)_{-1})$ are two already found consequent solutions for $\tilde{\varepsilon}_{-2}$ and $\tilde{\varepsilon}_{-1}$, respectively and ds is the arc-length step for which a new solution around the previous solution $(\tilde{u}(x)_{-2}, \tilde{v}(x)_{-2}, \tilde{\varepsilon}_{-2})$ along the arc-length of the solution branch is being sought. The corresponding bifurcation diagram is shown in Figure 6.2. We obtained a Andronov-Hopf bifurcation at $\varepsilon \approx 0.018497$ and a fold point at $\varepsilon \approx 0.95874$. This is considered as the reference bifurcation diagram. In 6.3, the contour plot of the $L2$ norm of the convergence of the solutions as computed with Finite Differences with respect to the size of the grid N as $L2_N = ||u_N - u_{2N}||_2$ is shown.

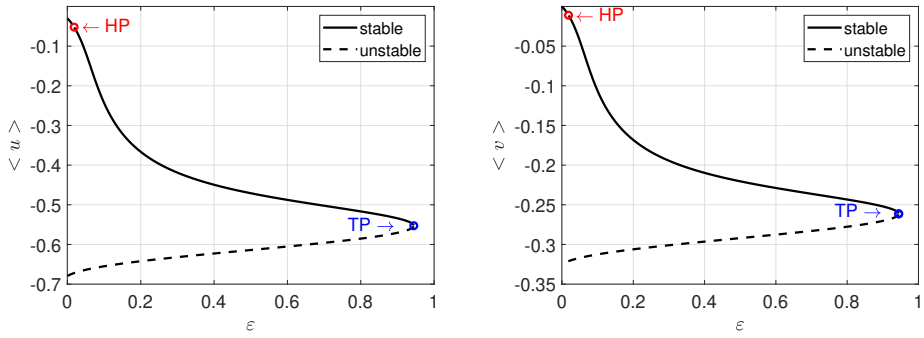


Fig. 6.2 Reference bifurcation diagram of the FHN PDEs with respect to ε as computed with FD and $N = 200$ points. (a) $< u >$, (b) $< v >$. Andronov-Hopf Point: HP_ε=0.01827931. Turning Point: TP_ε=0.94457768.

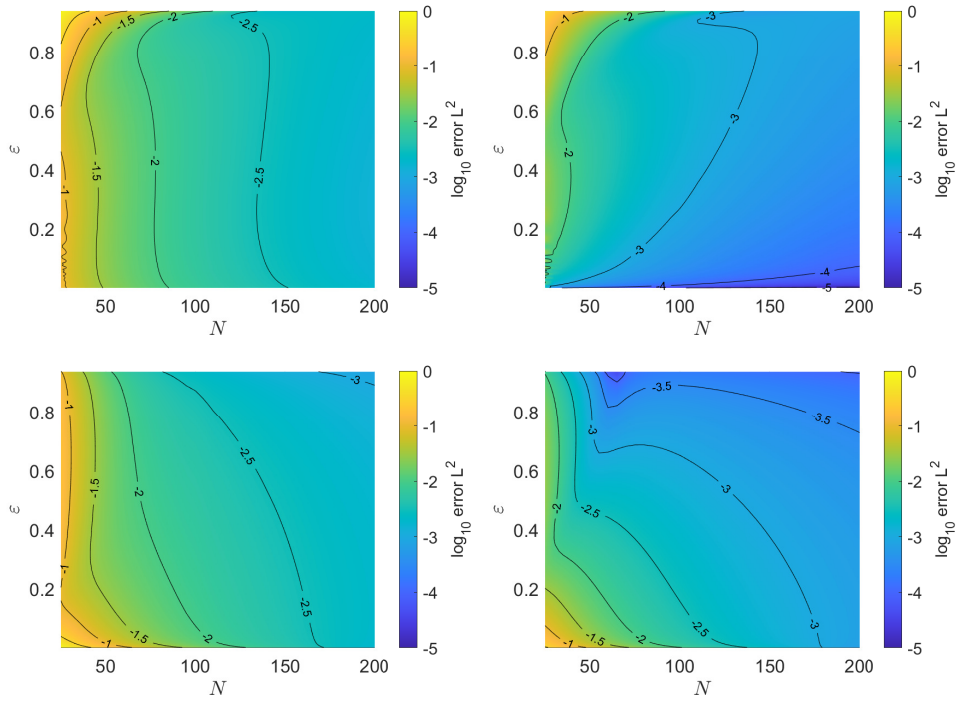


Fig. 6.3 Contour plot of the $L2$ norm of the convergence of the solutions as computed with Finite Differences with respect to the size of the grid N as $L2_N = ||u_N - u_{2N}||_2$. The convergence error was evaluated on 1001 grid points, using linear piecewise interpolation. (a) upper branch for u (b) upper branch for v (c) lower branch for u (d) lower branch for v.

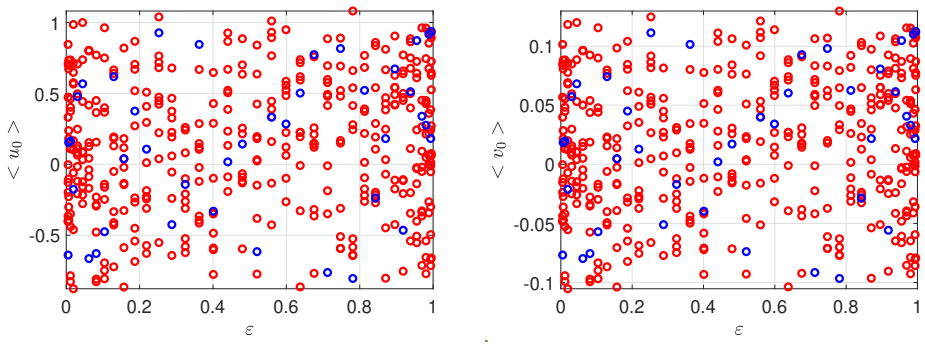


Fig. 6.4 Coarse initial conditions for (a) u and (b) v for the training. Every dot denotes a point whose ε and mean u (or v) were used for input data. Red dots are training points, blue points are testing points. The grid is spanned with Chebychev-Gauss-Lobatto points for epsilons in the interval $[0.005, 0.995]$ and the initial condition are randomly selected as in Eq. (6.12)

6.3 Numerical bifurcation analysis from microscopic simulations

For $u(x,t)$ and $v(x,t)$, we collected transients with a sampling rate of 1s, from 10 different random sampled initial conditions for 40 different values for the bifurcation parameter (ε). In particular, we created a grid of 40 different ε in [0005, 0.955] using Gauss-Chebychev-Lobatto points, while the 10 initial conditions are sampled according to Eq.6.12. Figure 6.4) depicts the total of 400 training initial conditions. Thus, we end up with a dataset consisting of 40 (values of ε)$\times$10 (initial conditions)$\times$448 (time points ignoring the first 2s of the transient)$\times$40 (space points) $\backsimeq$ 7.168.000 data points.

For learning the coarse-grained dynamics and construct the corresponding bifurcation diagram, we trained two FNNs (one for each one of the variables u and v).

The FNNs were constructed using two hidden layers with 12 units each. Hidden units were employed with the hyperbolic tangent sigmoid activation function while the regularization parameter was tuned and set $\lambda = 0.01$. For the training of the FNNs we used the Deep Learning toolbox of MATLAB 2021a on an Intel Core i5-8265U with up to 3.9 GHz frequency and a memory of 8 GB.

6.3.1 Numerical bifurcation analysis without feature selection

In Figures(6.5) we show the regression performance in the test dataset of the FNNs. Table (6.1) summarises the performance on the training and on test data set. The training phase of the FNN for $\hat{u}_t$ stopped at 828-th epoch and required around 4 hours with minimum tolerance set to $1e-07$. The training phase of the FNN for $\hat{v}_t$ stopped at 1000-th epoch and required around 4h with minimum tolerance set to $1e-10$. Differences between the predicted and the actual values of the time derivatives of u and v for three different values of ε are shown in Figure (6.6).

After training, we used the FNNs to compute with finite differences the quantities required for performing the bifurcation analysis (see Eq.(6.19)), i.e.:

$$\frac{\partial \hat{F}^u}{\partial u_j} = \frac{\hat{F}^u(u_j,v_j,\varepsilon) - \hat{F}^u(u_j+\delta,v_j,\varepsilon)}{2\delta}; \frac{\partial \hat{F}^u}{\partial v_j} = \frac{\hat{F}^u(u_j,v_j,\varepsilon) - \hat{F}^u(u_j,v_j+\delta,\varepsilon)}{2\delta}$$
$$\frac{\partial \hat{F}^v}{\partial u_j} = \frac{\hat{F}^v(u_j,v_j,\varepsilon) - \hat{F}^v(u_j+\delta,v_j,\varepsilon)}{2\delta}; \frac{\partial \hat{F}^v}{\partial v_j} = \frac{\hat{F}^v(u_j,v_j,\varepsilon) - \hat{F}^v(u_j,v_j+\delta,\varepsilon)}{2\delta}$$
$$\frac{\partial \hat{F}^u}{\partial \varepsilon} = \frac{\hat{F}^u(u_j,v_j,\varepsilon) - \hat{F}^u(u_j,v_j,\varepsilon+\delta)}{2\delta}; \frac{\partial \hat{F}^v}{\partial \varepsilon} = \frac{\hat{F}^v(u_j,v_j,\varepsilon) - \hat{F}^v(u_j,v_j,\varepsilon+\delta)}{2\delta},$$

with $\delta = 1e-06$. The reconstructed bifurcation diagrams are shown in Figure (6.9). Using the FNNs, we estimated the Andronov-Hopf point at $\varepsilon \approx 0.0191$ and the turning point at $\varepsilon \approx 0.9713$ We approximated the same points using the finite differences scheme in the previous section at $\varepsilon \approx 0.0183$ for the Andronov-Hopf point and at $\varepsilon \approx 0.9446$ for the turning point.

	test set				**train set**			
	MSE (u)	L_{inf} **(u)**	**MSE (v)**	L_{inf} **(v)**	**MSE (u)**	L_{inf} **(u)**	**MSE (v)**	L_{inf} **(v)**
FNN	7.90e-09	2.26e-02	1.56e-09	6.63e-03	1.31e-09	7.00e-03	2.78e-10	2.58e-03
FNN with FS	5.39e-08	2.93e-02	1.16e-08	7.65e-03	1.90e-08	2.64e-02	1.37e-09	4.70e-03

Table 6.1 Mean-square and L_{inf} errors between the predicted from the FNNs and the actual time derivatives of u and v.

6.3.2 Numerical bifurcation analysis with feature selection

Here, we aimed at identifying if any subset of features of the input domain can be used for parametrizing the intrinsic embeddings (and thus learn the RHS of the

Fig. 6.5 Regression results for the test dataset of the FNNs: (a) FNN without feature selection (R=0.99996) for $\hat{u}_t$, (b) FNN without feature selection (R=0.99992) for $\hat{v}_t$, (c) FNN with feature selection (R=0.99712) for $\hat{u}_t$. (d) FNN with feature selection (R=0.99950) for $\hat{v}_t$.

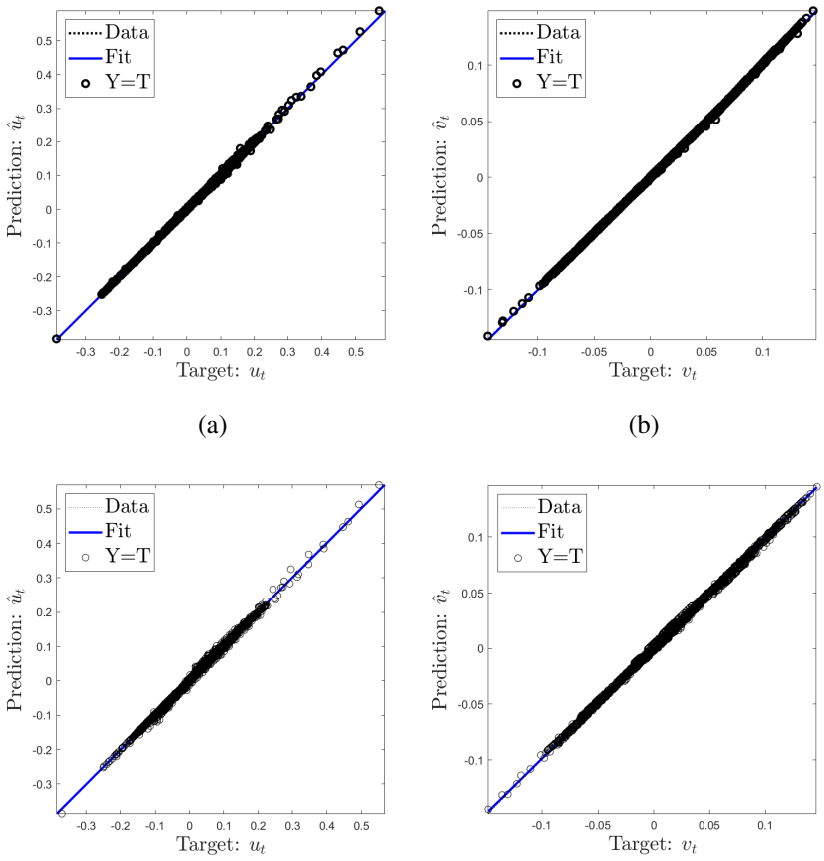
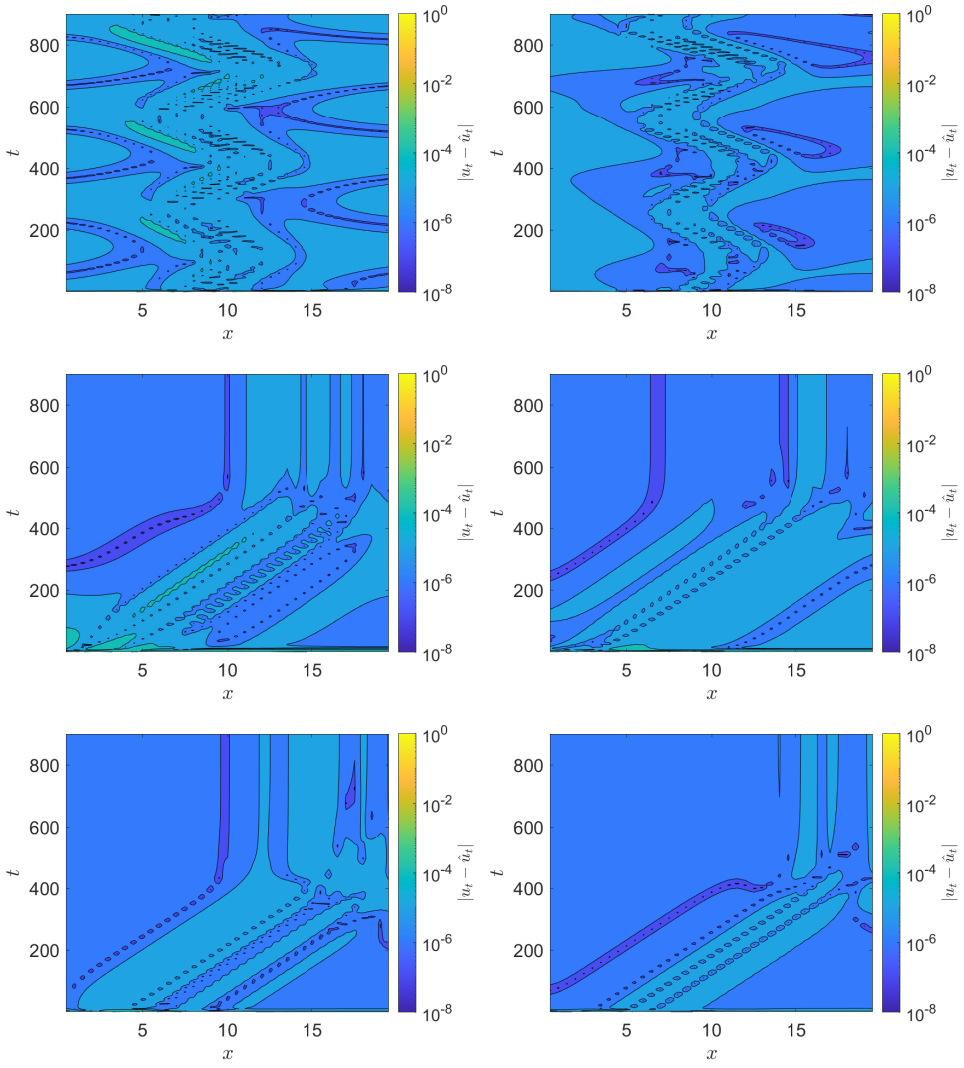


Fig. 6.6 Contour plot for $||u_t - \hat{u}_t||$ in the test dataset between the predicted time derivatives from the FNNs ($\hat{u}_t$) without feature selection and the actual time derivatives u_t ((a), (c), and (e)) and for $||v_t - \hat{v}_t||$ ((b), (d), and (f)) for different values of ε: (a) and (b) $\varepsilon = 0.0114$ near the Andronov-Hopf point, (c), (d) $\varepsilon = 0.4$, (e) and (f) $\varepsilon = 0.9383$ near the turning point.



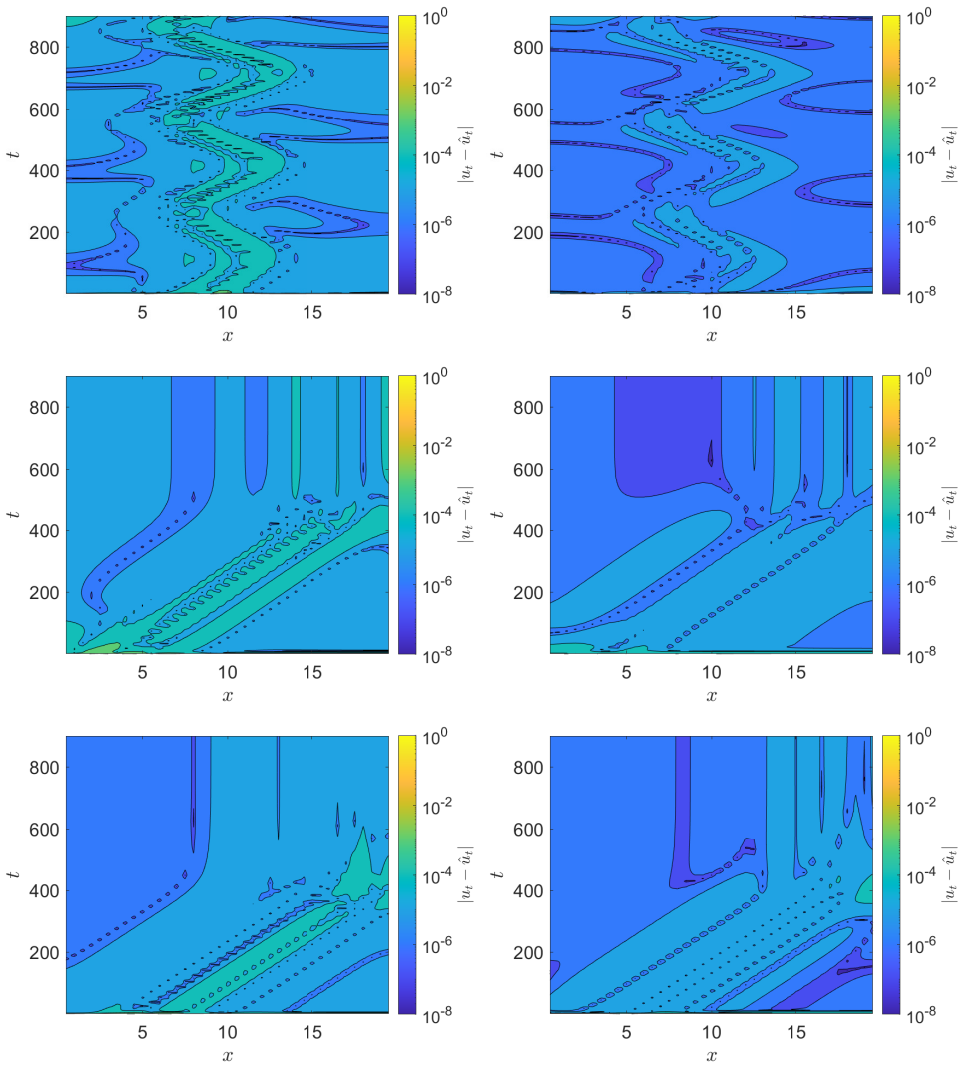


Fig. 6.7 Contour plot for $||u_t - \hat{u}_t||$ in the test dataset between the predicted time derivatives from the FNNs ($\hat{u}_t$) with feature selection and the actual time derivatives u_t ((a), (c), and (e)) and for $||v_t - \hat{v}_t||$ ((b), (d), and (f)) for different values of ε: (a) and (b) $\varepsilon = 0.0114$ near the Andronov-Hopf point, (c), (d) $\varepsilon = 0.4$, (e) and (f) $\varepsilon = 0.9383$ near the turning point.

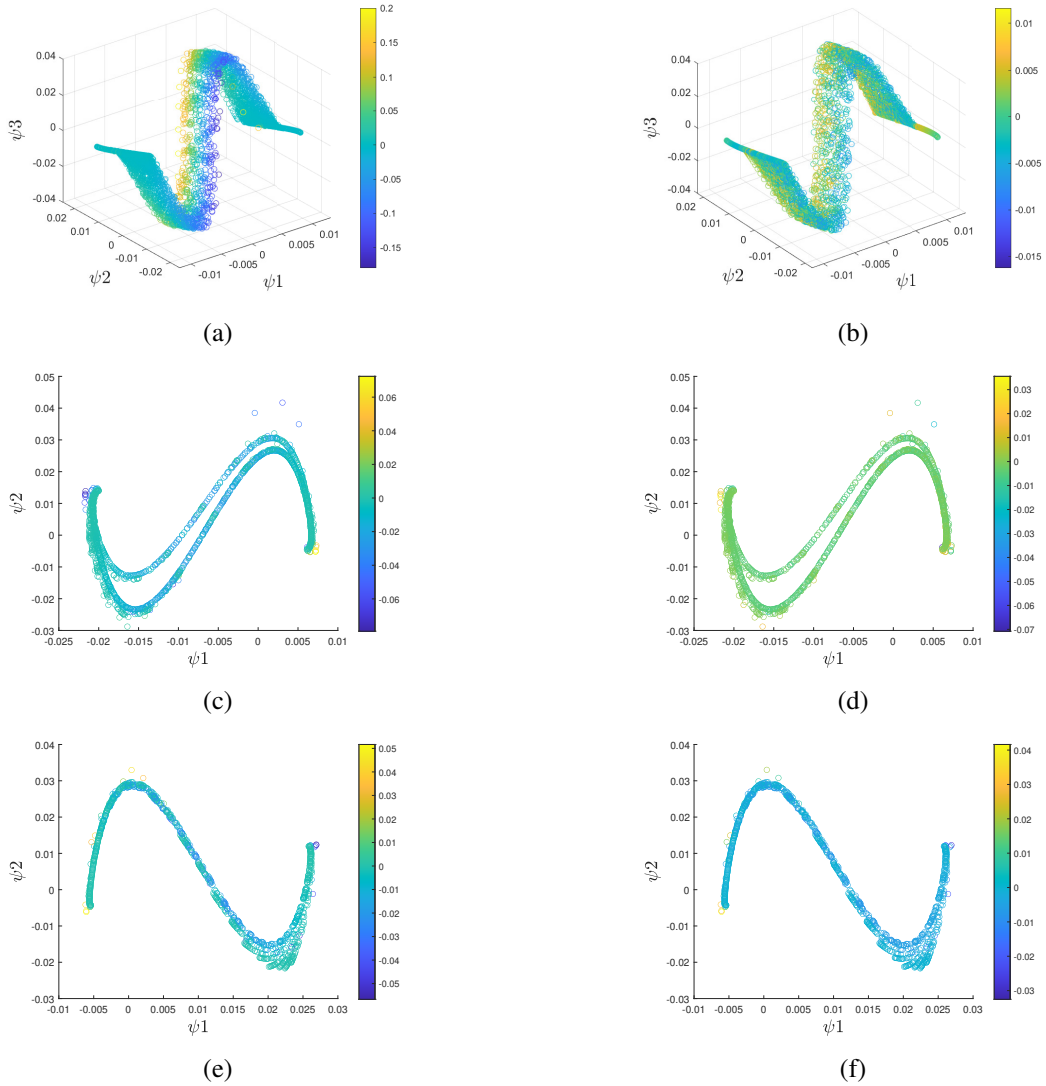
PDEs). We applied Diffusion Maps for every sample of the test dataset with the same value of the bifurcation parameter and identified the three parsimonious leading eigenvectors as described in section 4.3. We denote them as ϕ_1, ϕ_2, ϕ_3. We have set the width parameter of the Gaussian kernel to $\sigma = 10$. The three parsimonious Diffusion Maps coordinates for different values of the parameter epsilon are shown in figure (6.8). For $\varepsilon = 0.114$ that is close to the Andronov-Hopf point, the embedded space is a two dimensional "carpet" in the three dimensional space. The oscillatory behaviour leads to different values of the time derivative which can be effectively parametrized as shown by the coloring of the manifold (Figures 6.8a, 6.8b). On the contrary, for $\varepsilon = 0.4010$ and $\varepsilon = 0.9383$ where the solutions are the embedded space is one dimensional line in a two dimensional space since time derivatives converges rapidly to zero (Figures 6.8c,6.8e,6.8d and 6.8f). The "good" subsets of the input data domain are presented in Table (6.2). As expected, the best candidate features are the (u, v, u_{xx}) for u_t and (u, v, v_{xx}) for v_t which are the only features that indeed appear in the closed form of the FHN PDEs. For the selection of the three parsimonious Diffusion Maps coordinates, we used the datafold package in python ([201]).

	$u_t = (\phi_1^u, \phi_2^u, \phi_3^u)$		$v_t = (\phi_1^v, \phi_2^v, \phi_3^v)$	
	Features	**Sum Total Losses**	**Features**	**Sum Total Losses**
1d	(u)	4.3E-03	(u)	7.6E-03
2d	(u, v)	6.37E-06	(u, v)	1.91E-05
3d	(u, v, u_{xx})	2.77E-07	(u, v, v_{xx})	6.29E-07
4d	(u, v, u_x, u_{xx})	1.03E-07	(u, v, v_x, v_{xx})	1.34E-07

Table 6.2 The "best" variables that can effectively parametrize the intrinsic coordinates ($(\phi_1^u, \phi_2^2, \phi_3^u)$ and $(\phi_1^v, \phi_2^v, \phi_3^v)$) and the corresponding sums of total loses across all the values of the bifurcation parameter ε.

Finally, we repeated the same steps but now using as inputs in the FNNs the reduced input domain as obtained from the feature selection process, the bifurcation parameter. Figures (6.5c) and (6.5d) show the regression results of the schemes. Table (6.1) summarizes the performance of the schemes on the training and the test set. The training phase of the FNN for $\hat{u}_t$ stoped at 1000-th epoch and required 3 hours and 37 minutes with minimum tolerance set to $1e-07$. The training phase of the FNN for $\hat{v}_t$ stoped at 1000-th epoch and required 4 hours and 19 minutes. Figure (6.7) illustrates the differences between the predicted from the FNNs with feature

Fig. 6.8 (a) and (b): The three parsimonious Diffusion Maps coordinates for $\varepsilon = 0.01114$ near the Andronov-Hopf point. In this region the solutions are oscillating. (c) and (d): The two parsimonious Diffusion maps coordinates for $\varepsilon = 0.4010$. (e) and (f): The two parsimonious Diffusion maps coordinates for $\varepsilon = 0.9383$ near the turning point. Colors represent u_t ((a), (c), (e)) and v_t ((b), (d), (f)).



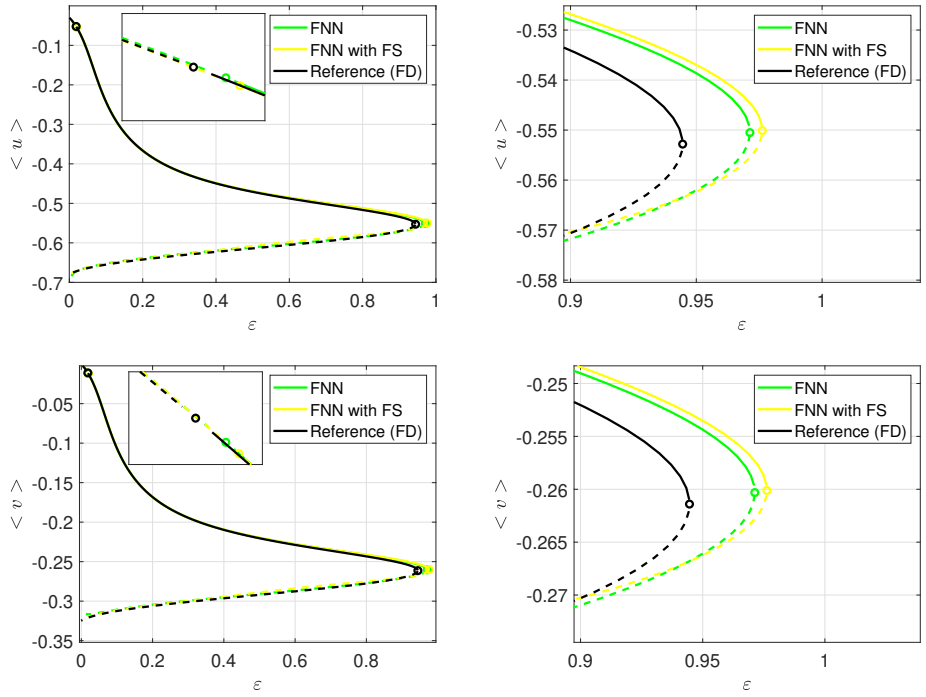


Fig. 6.9 Reconstructed bifurcation diagram from the Lattice-Boltzmann simulations of the FHN dynamics with respect to ε with FNNs with feature selection. (a) $< u >$; the inset zooms near the Andronov-Hopf bifurcation point (b) $< u >$ the inset zooms near the turning Point, (c) $< v >$; the inset zooms near the Andronov-Hopf bifurcation point, (d) $< v >$; the inset zooms near the turning Point.

selection and the actual time derivatives. Hence, we have:

$$\begin{aligned} \frac{\partial u}{\partial t} &= \hat{F}^u(u, v, u_{xx}, \varepsilon), \\ \frac{\partial v}{\partial t} &= \hat{F}^v(u, v, v_{xx}, \varepsilon) \end{aligned} \tag{6.20}$$

where $\hat{F}^u$ and $\hat{F}^v$ are the outputs of the FNNs and we perform Arc Length for ε, computing numerically the Jacobians of F^u and F^v . Thus, the data-driven bifurcation diagram with feature selection is shown in Figure 6.9. Now, using the FNNs, we estimated the Andronov-Hopf point at $\varepsilon \approx 0.0195$ and the turning point at $\varepsilon \approx 0.9762$.

6.4 Discussion

Buliding on previous efforts [5], we present a machine-learning methodology for constructing the coarse-bifurcation diagrams using fine scale simulations in space and time. The proposed approach is a three tier one. In the first step, we use non-linear manifold-learning and in particular Diffusion Maps to select the right coarse-scale observables that define the low-dimensional manifold on which the emergent dynamics evolve in the parameter space. At the second step, we learn the right-hand-side of the effective PDEs with respect to the coarse-scale observables; here we used FNNs. Finally, based on the constructed black-box model, we construct the coarse-grained bifurcation diagrams exploting the arsenal of numerical bifurcation. To demonstrate the approach, we used Lattice-Boltzmann simulations of the FitzHugh-Nagumo PDEs and compared the obtained bifurcation diagram with the one constructed directly by discretizing the PDEs with Finite-Differences.

The results show that the proposed machine-learning framework was able to identify the correct variables that are required to model the emergent dynamics and based on them to construct accurately the actual bifurcation diagram. In terms of approximation accuracy of the observed dynamics, both approches (with and without feature selection) contsructed the bifurcation diagram with high accuracy.

Here, we have focused on constructing black-box models of PDEs. The approach can be extended to construct “gray-box" models by incorporating information from the physics into the machine learning architecture [202]. Futhermore, based on previous efforts aiming at extracting normal forms of ODEs from data [203], the approach can be extended to discover normal forms of PDEs.

Chapter 7

Conclusions

In this Thesis, we developed and implemented physics-informed data-driven computational tools, based on numerical analysis and machine learning to confront with the problem of coarse-grained modelling and numerical bifurcation analysis of the emergent dynamics of complex systems. We considered three different open problems in the field, namely that of (i) solving the inverse source localization problem with a focus in computational neuroscience and neuroimaging; here we compared the performance of well-established regularization techniques on real EEG data, thus exploiting the information from metanalytics studies from fMRI data, (ii) developing a machine learning method for solving the inverse problem in complex systems modelling, i.e. identifying an appropriate set of observables that parametrize low-dimensional manifolds where the emergent dynamics evolve and based on them construct models in the form of PDEs for systematically study the dynamics with the aid of numerical bifurcation analysis algorithms, and (iii) developing a machine-learning method for solving the direct problem, namely that of the numerical solution of low-dimensional initial value problems of ODEs with a focus on stiff-problems containing sharp gradients.

For the first problem, we revealed extra evidence about the crucial role of the parietal lobules in working memory tasks in children with controlled epilepsy [111]. This finding was actually obtained by all regularization methods that we used, thus indicating that it is robust enough and may serve as “biomarker" for identifying, monitoring and assessing epilepsy in children. This is the first time that such a com-

parative analysis for the specific important problem has been done. In the present thesis, we used EEG signals coming from a relevant small number of electrodes, something that may affect the accuracy of the results. Investigating if similar results would be revealed with more dense EEG recordings would be of paramount importance. We approached the source localization problem from the Tikhonov regularization point of view. There have been proposed different approaches for obtaining the sources that create the EEG signals such as beamforming techniques, which independently scan for dipoles within a grid containing candidate locations (i.e. source points) [204, 205]. The goal is to estimate the activity at a source point or region while avoiding the crosstalk from other regions so that these affect as little as possible the estimate at the region of interest. Recently, there have been efforts to employ machine learning for solving the source localization problem [206, 207]. The idea of this promising approach is to utilize machine learning methodologies that have already applied for solving inverse imaging problems such as X-ray computed tomography (CT) [208, 209], MRI [210, 211], image super resolution [212, 213], seismic tomography [214] and more. Since the source localization problem constitutes a very sensitive problem in many ways, from the noise of the EEG recordings to the effect of the head model and the approximation error of the applied algorithm, comparison between different approaches and validation of the results could help to investigate the weaknesses. Moving from the sensor space into the source space leads the scientific community to various new open issues. Except from the obvious result of the visualization of the true regions that are involved, ESL is being used for more scientific purposes. Studying the functional connectivity of human brain using source time series instead of the measured EEG scalp recordings is indisputable one of the most widely studied topics in computational neuroscience [215–219]. Another of utmost, source connectivity is a topic with many open issues and various of challenges and many optimism results that turn a lot of researchers to deal with them. It is of great interest that we can continue our analysis, implementing functional connectivity methods such as Granger causality [220], Partial Directed Coherence (PDC) [221] and others and construct the intra-networks of causality that govern the multiple source time series.

For the second problem, building on previous efforts [5], we developed a data-driven computational framework for constructing coarse-bifurcation diagrams from fine scale observations without assuming any knowledge for a closed form model [95]. For our illustrations, we have chosen to learn and reconstruct the dynamics

of the two coupled Fitzhugh-Nagumo PDEs, a celebrated model in the field of mathematical Neuroscience. The proposed methodology succeeded to effectively identify the set coarse variables that are required to model the emergent dynamics and based on them approximate with high accuracy the actual bifurcation diagram. For this purpose, we used a feature selection method based on the Diffusion Maps Algorithm. To the best of our knowledge, this is the first time that such a methodology has be addressed for constructing bifurcation diagrams from data. During the last decades, there have been developed many feature selection algorithms such as group LASSO [222], random forests [223], recursive feature elimination [224] and more. It would be interesting to check if other methods would suggest different set of features for describing the coarse-grained dynamics. Lovelett et. al. [202] incorporated physics information into the machine learning architecture and constructed "grey-box" models which would be a promising expansion of the presented methodology. Another future direction could be the generalization for extracting normal form of PDEs from data as in [203]. As regard the task of learning the dynamics, we applied FNNs. FNNs have high computationally cost and demand many hours when handling with large datasets. Evaluation of RPNNs/ ELMs may reduce dramatically the computational cost of the methodology. We constructed fine-scale observations from the LBM microsimulator for the FHN equations. However, this framework should be validated in observations coming from real systems. In this case, methods for dealing with possible noisy or extreme values should be integrated [225]. Finally, towards the construction of bifurcation diagrams, it would be interesting to exploit and compare the performance of other schemes such as the one addressed in Wang et al. [226], a physics-informed machine learning scheme based on deep learning to learn the solution operator of arbitrary PDEs and that in Kovachki et al. [227] where it has been addressed the concept of Neural Operators, which are mesh-free, infinitedimensional operators with neural networks to learn surrogate functional maps for the solution operators of PDEs.

Finally for the third problem, we expoited the high speed and approximation properties of RPNNs for the efficient numerical solution of low-dimensional systems of stiff ODEs with encouraging results. In particular, results show that the proposed method in some cases especially in the regions where sharp gradients arise outperform the built in matlab function `ode15s` reagarding the numerical accuracy. The proposed numerical scheme required in general more time than matlab's functions but it was fairly comparable. An interesting feature of the proposed method is

that provides an analytical expression for the approximation of the solution upon training with a finite set of collocation points. This feature leads to less computational cost in "dense" sets of points. Definitely, an optimized code may leads to smaller computational times. A systematic validation of its performance over various suggested benchmark problems ([228, 229]) is a possible subject of future work. The algorithm was evaluated in low-dimensional systems of ODEs consisting of up to three equations. In many real life complex systems of PDEs, one may ends up to a stiff ODE system of much more equations. A possible future expansion of the proposed methodology applied to higher order systems is one of our future tasks. Since our method computes the solution in the whole domain, the issue for selecting the number of collocation points needed for extracting a relevant solution needs further investigation. Applying an adaptive expansion, which would divide the whole interval into subintervals accordingly to the approximation error, may overcome this issue. Furthermore, a systematic tuning of the hyperparameters of the algorithm must be done. It is interesting to investigate if different architectures or different set of hyperparameters could amplify the approximation accuracy. Another useful procedure would be the stability and error propagation analysis. The convergence properties, the stability and the theoretical foundations of the method need to be learnt in detail in the future. Finally, a comparison with other ML schemes [86, 88, 182, 230, 231] that have been proposed for obtaining the solutions of differential equations is important. Since these techniques have emerged only recently, the exhaustive comparison between the various schemes could reveal future directions and/or limitations of this approach.

The common conclusion from the problems we dealt with in this Thesis is that the synergistic integration between machine learning or numerical analysis algorithms and physical models opens new horizons and at the same moment poses more challenges in scientific computing. Looking to the future, where more than a trillion sensors (airborne, seaborne, satellite remote sensing and others) are expected to provide us with a wealth of multi-fidelity observations, the effective integration of the physical knowledge to the traditional machine leaning and numerical analysis algorithms will come to play a decisive role in our ambition to understand the world around us.

References

[1] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[2] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85, 2009.

[3] Ioannis G. Kevrekidis, C. William Gear, James M. Hyman, Panagiotis G. Kevrekidis, Olof Runborg, and Constantinos Theodoropoulos. Equation-free, coarse-grained multiscale computation: Enabling mocroscopic simulators to perform system-level analysis. *Communications in Mathematical Sciences*, 1(4):715–762, 2003.

[4] Ioannis G. Kevrekidis, C. William Gear, and Gerhard Hummer. Equation-free: The computer-aided analysis of complex multiscale systems. *AIChE Journal*, 50(7):1346–1355, 2004.

[5] Seungjoon Lee, Mahdi Kooshkbaghi, Konstantinos Spiliotis, Constantinos I. Siettos, and Ioannis G. Kevrekidis. Coarse-scale pdes from fine-scale observations via machine learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(1):013141, 2020.

[6] Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006. Special Issue: Diffusion Maps and Wavelets.

[7] Robert Gerstberger and Peter Rentrop. Feedforward neural nets as discretization schemes for ODEs and DAEs. *Journal of Computational and Applied Mathematics*, 82(1-2):117–128, 1997.

[8] Isaac E. Lagaris, Aristidis Likas, and Dimitrios I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 1998.

[9] Hyuk Lee and In Seok Kang. Neural algorithm for solving differential equations. *Journal of Computational Physics*, 91(1):110–131, 1990.

[10] Andrew J. Meade Jr and Alvaro A. Fernandez. The numerical solution of linear ordinary differential equations by feedforward neural networks. *Mathematical and Computer Modelling*, 19(12):1–25, 1994.

[11] Lawrence F Shampine and Charles William Gear. A user's view of solving stiff ordinary differential equations. *SIAM review*, 21(1):1–17, 1979.

[12] J.D. Lambert. *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem*. Wiley, 1992.

[13] Mario De Florio, Enrico Schiassi, Barry D Ganapol, and Roberto Furfaro. Physics-informed neural networks for rarefied-gas dynamics: Thermal creep flow in the bhatnagar–gross–krook approximation. *Physics of Fluids*, 33(4):047110, 2021.

[14] Ioannis Th. Famelis and Vasiliki Kaloutsa. Parameterized neural network training for the solution of a class of stiff initial value systems. *Neural Computing and Applications*, 33(8):3363–3370, 2021.

[15] Dimitris A Goussis and Mauro Valorani. An efficient iterative algorithm for the approximation of the fast and slow dynamics of stiff systems. *Journal of Computational Physics*, 214(1):316–346, 2006.

[16] M Hadjinicolaou and Dimitris A Goussis. Asymptotic solution of stiff pdes with the csp method: the reaction diffusion equation. *SIAM Journal on Scientific Computing*, 20(3):781–810, 1998.

[17] Weiqi Ji, Weilun Qiu, Zhiyu Shi, Shaowu Pan, and Sili Deng. Stiff-pinn: Physics-informed neural network for stiff chemical kinetics. *The Journal of Physical Chemistry A*, 125(36):8098–8106, 2021.

[18] Mauro Valorani and Dimitrios A Goussis. Explicit time-scale splitting algorithm for stiff problems: auto-ignition of gaseous mixtures behind a steady shock. *Journal of Computational Physics*, 169(1):44–79, 2001.

[19] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Inferring solutions of differential equations using noisy multi-fidelity data. *Journal of Computational Physics*, 335:736–746, 2017.

[20] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[21] Francesco Calabrò, Gianluca Fabiani, and Constantinos Siettos. Extreme learning machine collocation for the numerical solution of elliptic PDEs with sharp gradients. *Computer Methods in Applied Mechanics and Engineering*, 387:114188, 2021.

[22] Gianluca Fabiani, Francesco Calabrò, Lucia Russo, and Constantinos Siettos. Numerical solution and bifurcation analysis of nonlinear partial differential equations with extreme learning machines. *Journal of Scientific Computing*, 89:44, 2021.

[23] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yan. Physics-informed machine learning. *Nature Reviews Physics*, 3:422–440, 2021.

[24] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, and Prabhat. Deep learning and process understanding for data-driven earth system science. *Nature*, 566:195–204, 2019.

[25] Mark S. Alber, Adrian Buganza Tepole, William R. Cannon, Suvranu De, Salvador Dura-Bernal, Krishna C. Garikipati, George Em Karniadakis, William W. Lytton, Paris Perdikaris, Linda R. Petzold, and Ellen Kuhl. Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. *NPJ Digital Medicine*, 2, 2019.

[26] J. Bielecki, L. Giacomelli, V. Kiptily, M. Scholz, K. Drozdowicz, S. Conroy, T. Craciunescu, and M. Kempenaars. Phillips-tikhonov regularization with a priori information for neutron emission tomographic reconstruction on joint european torus. *Review of Scientific Instruments*, 86(9):093505, 2015.

[27] D Shchekotin and V Sizikov. Using regularization and a priori information to enhance tomographic image quality. *The Imaging Science Journal*, 54(3):147–151, 2006.

[28] Christophe Phillips, Michael D. Rugg, and Karl J. Friston. Anatomically informed basis functions for eeg source localization: Combining functional and anatomical constraints. *NeuroImage*, 16(3, Part A):678–695, 2002.

[29] Munsif Ali Jatoi, Nidal Kamel, Aamir Saeed Malik, Ibrahima Faye, and Tahamina Begum. A survey of methods used for source localization using eeg signals. *Biomedical Signal Processing and Control*, 11:42 – 52, 2014.

[30] Kari Modalsli Aaberg, Nina Gunnes, Inger Johanne Bakken, Camilla Lund Søraas, Aleksander Berntsen, Per Magnus, Morten I. Lossius, Camilla Stoltenberg, Richard Chin, and Pål Surén. Incidence and prevalence of childhood epilepsy: A nationwide cohort study. *Pediatrics*, 139(5):e20163908, April 2017.

[31] Sharon Davies, Isobel Heyman, and Robert Goodman. A population survey of mental health problems in children with epilepsy. *Developmental Medicine & Child Neurology*, 45(05), April 2003.

[32] Federico E. Turkheimer, Fernando E. Rosas, Ottavia Dipasquale, Daniel Martins, Erik D. Fagerholm, Paul Expert, František Váša, Louis-David Lord, and Robert Leech. A complex systems perspective on neuroimaging studies

of behavior and its disorders. *The Neuroscientist*, 0(0):1073858421994784, 0. PMID: 33593120.

[33] Karl J. Friston. The free-energy principle: a rough guide to the brain? *Trends in Cognitive Sciences*, 13:293–301, 2009.

[34] Olaf Sporns. The non-random brain: Efficiency, economy, and complex dynamics. *Frontiers in Computational Neuroscience*, 5, 2011.

[35] Edward Bullmore, Anna Barnes, Danielle Bassett, Alex Fornito, Manfred Kitzbichler, David Meunier, and John Suckling. Generic aspects of complexity in brain imaging data and other biological systems. *NeuroImage*, 47:1125–34, 06 2009.

[36] Gustavo Deco, Viktor Jirsa, and Anthony McIntosh. Emerging concepts for the dynamical organization of resting-state activity in the brain. *Nature reviews. Neuroscience*, 12:43–56, 01 2011.

[37] Danielle Bassett and Michael Gazzaniga. Understanding complexity in the human brain. *Trends in cognitive sciences*, 15:200–9, 05 2011.

[38] Qawi Telesford, Sean Simpson, Jonathan Burdette, Satoru Hayasaka, and Paul Laurienti. The brain as a complex system: Using network science as a tool for understanding the brain. *Brain connectivity*, 1:295–308, 10 2011.

[39] Douglas S. Goodin, Michael J. Aminoff, and Kenneth Laxer. Detection of epileptiform activity by different noninvasive eeg methods in complex partial epilepsy. *Annals of neurology*, 27:330–4, 03 1990.

[40] S J M Smith. Eeg in the diagnosis, classification, and management of patients with epilepsy. *Journal of Neurology, Neurosurgery and Psychiatry*, 76(suppl 2):ii2–ii7, 2005.

[41] Mohammed M. S. Jan, Mark Sadler, and Susan R. Rahey. Lateralized postictal eeg delta predicts the side of seizure surgery in temporal lobe epilepsy. *Epilepsia*, 42(3):402–405, 2001.

[42] Christina M. Krause, Petra-Ann Boman, Lauri Sillanmäki, Tarja Varho, and Irma E. Holopainen. Brain oscillatory eeg event-related desynchronization (erd) and -sychronization (ers) responses during an auditory memory task are altered in children with epilepsy. *Seizure*, 17(1):1 – 10, 2008.

[43] Ivan Myatchin, Maarten Mennes, Heidi Wouters, Peter Stiers, and Lieven Lagae. Working memory in children with epilepsy: An event-related potentials study. *Epilepsy Research*, 86:183–190, 2009.

[44] Foteini Protopapa, Constantinos I. Siettos, Ioannis Evdokimidis, and Nikolaos Smyrnis. Granger causality analysis reveals distinct spatio-temporal connectivity patterns in motor and perceptual visuo-spatial working memory. *Frontiers in Computational Neuroscience*, 8:146, 2014.

[45] Li Yang, Irina Shklyar, Hyang Woon Lee, Celestine C. Ezeani, Joseph Anaya, Samantha Balakirsky, Xiao Han, Sheila Enamandram, Clara Men, Joyce Y. Cheng, Abigail Nunn, Tanya Mayer, Czestochowa Francois, Molly Albrecht, Alan L. Hutchison, Ee-Lynn Yap, Kevin Ing, Gvantsa Didebulidze, Bo Xiao, Hamada Hamid, Pue Farooque, Kamil Detyniecki, Joseph T. Giacino, and Hal Blumenfeld. Impaired consciousness in epilepsy investigated by a prospective responsiveness in epilepsy scale (res). *Epilepsia*, 53(3):437–447, 2012.

[46] Eric van Diessen, Willem M. Otte, Kees P. J. Braun, Cornelis J. Stam, and Floor E. Jansen. Improved diagnosis in children with partial epilepsy using a multivariable prediction model based on eeg network characteristics. *PLOS ONE*, 8(4):1–8, 04 2013.

[47] Roberta Grech, Tracey Cassar, Joseph Muscat, Kenneth P. Camilleri, Simon G. Fabri, Michalis Zervakis, Petros Xanthopoulos, Vangelis Sakkalis, and Bart Vanrumste. Review on solving the inverse problem in eeg source analysis. *Journal of NeuroEngineering and Rehabilitation*, 5(1):25, Nov 2008.

[48] Roberto D. Pascual-Marqui. Review of methods for solving the eeg inverse problem. 1999.

[49] Chris Plummer, A. Simon Harvey, and Mark Cook. EEG source localization in focal epilepsy: Where are we now? *Epilepsia*, 49(2):201–218, February 2008.

[50] Pierre Mégevand and Margitta Seeck. Electroencephalography, magnetoencephalography and source localization. *Current Opinion in Neurology*, 31(2):176–183, April 2018.

[51] Kitti Kaiboriboon, Hans O. Lüders, Mehdi Hamaneh, John Turnbull, and Samden D. Lhatoo. EEG source imaging in epilepsy—practicalities and pitfalls. *Nature Reviews Neurology*, 8(9):498–507, August 2012.

[52] John S. Ebersole and Susan M. Ebersole. Combining MEG and EEG source modeling in epilepsy evaluations. *Journal of Clinical Neurophysiology*, 27(6):360–371, December 2010.

[53] Friederike Moeller, Ulrich Stephani, and Michael Siniatchkin. Simultaneous EEG and fMRI recordings (EEG-fMRI) in children with epilepsy. *Epilepsia*, 54(6):971–982, May 2013.

[54] Munsif Ali Jatoi, Nidal Kamel, Aamir Saeed Malik, and Ibrahima Faye. Eeg based brain source localization comparison of sloreta and eloreta. *Australasian Physical & Engineering Sciences in Medicine*, 37(4):713–721, Dec 2014.

[55] F. Cincotti, C. Babiloni, C. Miniussi, F. Carducci, D. Moretti, S. Salinari, R. Pascual-Marqui, P. M. Rossini, and F. Babiloni. EEG deblurring techniques in a clinical context. *Methods of Information in Medicine*, 43(01):114–117, 2004.

[56] Jun Yao and Julius P.A. Dewald. Evaluation of different cortical source localization methods using simulated and experimental EEG data. *NeuroImage*, 25(2):369–382, apr 2005.

[57] Yohan Attal and Denis Schwartz. Assessment of subcortical source localization using deep brain activity imaging model with minimum norm operators: A MEG study. *PLoS ONE*, 8(3):e59856, mar 2013.

[58] Anett Seeland, Mario M. Krell, Sirko Straube, and Elsa A. Kirchner. Empirical comparison of distributed source localization methods for single-trial detection of movement preparation. *Frontiers in Human Neuroscience*, 12, sep 2018.

[59] Azeez Adebimpe, Ardalan Aarabi, Emilie Bourel-Ponchel, Mahdi Mahmoudzadeh, and Fabrice Wallois. EEG resting state functional connectivity analysis in children with benign epilepsy with centrotemporal spikes. *Frontiers in Neuroscience*, 10, March 2016.

[60] Kristina Groening, Verena Brodbeck, Friederike Moeller, Stephan Wolff, Andreas van Baalen, Christoph M. Michel, Olav Jansen, Rainer Boor, Gert Wiegand, Ulrich Stephani, and Michael Siniatchkin. Combination of EEG–fMRI and EEG source analysis improves interpretation of spike-associated activation networks in paediatric pharmacoresistant focal epilepsies. *NeuroImage*, 46(3):827–833, July 2009.

[61] Lydia Elshoff, Kristina Groening, Frédéric Grouiller, Gert Wiegand, Stephan Wolff, Christoph Michel, Ulrich Stephani, and Michael Siniatchkin. The value of EEG-fMRI and EEG source analysis in the presurgical setup of children with refractory focal epilepsy. *Epilepsia*, 53(9):1597–1606, July 2012.

[62] Andreas A. Ioannides, Lichan Liu, Vahe Poghosyan, Khalid Hamandi, and George K. Kostopoulos. Source-estimation from non-invasive recordings of brain electrical activity in sleep and epilepsy. In *Cyberphysical Systems for Epilepsy and Related Brain Disorders*, pages 61–86. Springer International Publishing, 2015.

[63] R.A. Chowdhury, I. Merlet, G. Birot, E. Kobayashi, A. Nica, A. Biraben, F. Wendling, J.M. Lina, L. Albera, and C. Grova. Complex patterns of spatially extended generators of epileptic activity: Comparison of source localization methods cMEM and 4-ExSo-MUSIC on high resolution EEG and MEG data. *NeuroImage*, 143:175–195, dec 2016.

[64] Mahmoud Hassan, Isabelle Merlet, Ahmad Mheich, Aya Kabbara, Arnaud Biraben, Anca Nica, and Fabrice Wendling. Identification of interictal epileptic networks from dense-EEG. *Brain Topography*, 30(1):60–76, aug 2016.

[65] P Fergus, David Hignett, Abir Hussain, Dhiya Al-Jumeily, and Khaled Abdel-Aziz. Automatic epileptic seizure detection using scalp eeg and advanced artificial intelligence techniques. *BioMed Research International*, 2015:17, 01 2015.

[66] J.L. Hudson, M. Kube, R.A. Adomaitis, I.G. Kevrekidis, A.S. Lapedes, and R.M. Farber. Nonlinear signal processing and system identification: applications to time series from electrochemical reactions. *Chemical Engineering Science*, 45(8):2075–2081, 1990.

[67] K. Krischer, R. Rico-Martinez, I. G. Kevrekidis, HH Rotermund, G. Ertl, and JL Hudson. Model identification of a spatiotemporally varying catalytic reaction. *Aiche JournalAiche Journal*, 39(1):89–98, JAN 1993 1993.

[68] R. Rico-Martinez, J.S. Anderson, and I.G. Kevrekidis. Continuous-time nonlinear signal processing: a neural network based approach for gray box identification. In *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, pages 596–605, 1994.

[69] J.S. Anderson, I.G. Kevrekidis, and R. Rico-Martinez. A comparison of recurrent training algorithms for time series analysis and system identification. *Computers & Chemical Engineering*, 20:S751–S756, 1996. European Symposium on Computer Aided Process Engineering-6.

[70] R. González-García, R. Rico-Martínez, and I.G. Kevrekidis. Identification of distributed parameter systems: A neural net based approach. *Computers & Chemical Engineering*, 22:S965–S968, 1998. European Symposium on Computer Aided Process Engineering-8.

[71] AP Alexandridis, CI Siettos, HK Sarimveis, AG Boudouvis, and GV Bafas. Modelling of nonlinear process dynamics using kohonen's neural networks, fuzzy systems and chebyshev series. *Computers & Chemical Engineering*, 26(4-5):479–486, 2002.

[72] Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007.

[73] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[74] Alexei G. Makeev, Dimitrios Maroudas, and Ioannis G. Kevrekidis. "coarse" stability and bifurcation analysis using stochastic simulators: Kinetic monte carlo examples. *The Journal of Chemical Physics*, 116(23):10083–10091, June 2002.

[75] C. I. Siettos, C. W. Gear, and I. G. Kevrekidis. An equation-free approach to agent-based computation: Bifurcation analysis and control of stationary states. *EPL (Europhysics Letters)*, 99(4):48007, August 2012.

[76] Christophe Vandekerckhove, Ioannis Kevrekidis, and Dirk Roose. An efficient newton-krylov implementation of the constrained runs scheme for initializing on a slow manifold. *Journal of Scientific Computing*, 39(2):167–188, 2009.

[77] Giovanni Samaey, Wim Vanroose, Dirk Roose, and Ioannis G Kevrekidis. Newton–krylov solvers for the equation-free computation of coarse traveling waves. *Computer Methods in Applied Mechanics and Engineering*, 197(43-44):3480–3491, 2008.

[78] Giovanni Samaey and Wim Vanroose. An analysis of equivalent operator preconditioning for equation-free newton–krylov methods. *SIAM journal on numerical analysis*, 48(2):633–658, 2010.

[79] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences*, 102(21):7426–7431, 2005.

[80] Boaz Nadler, Stéphane Lafon, Ronald R Coifman, and Ioannis G Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Applied and Computational Harmonic Analysis*, 21(1):113–127, 2006.

[81] Amit Singer, Radek Erban, Ioannis G Kevrekidis, and Ronald R Coifman. Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps. *Proceedings of the National Academy of Sciences*, 106(38):16090–16095, 2009.

[82] Boaz Nadler, Stéphane Lafon, Ronald R. Coifman, and Ioannis G. Kevrekidis. Diffusion maps - a probabilistic interpretation for spectral embedding and clustering algorithms. 2008.

[83] Seungjoon Lee, Ioannis G Kevrekidis, and George Em Karniadakis. A resilient and efficient cfd framework: Statistical learning tools for multi-fidelity and heterogeneous information fusion. *Journal of Computational Physics*, 344:516–533, 2017.

[84] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Machine learning of linear differential equations using gaussian processes. *Journal of Computational Physics*, 348:683–693, 2017.

[85] Neha Yadav, Anupam Yadav, Manoj Kumar, et al. *An introduction to neural network methods for differential equations*. Springer, 2015.

[86] Suchuan Dong and Zongwei Li. Local extreme learning machines and domain decomposition for solving linear and nonlinear partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 387:114129, 2021.

[87] Tamirat Temesgen Dufera. Deep neural network for system of ordinary differential equations: vectorized algorithm and simulation. *Machine Learning with Applications*, page 100058, 2021.

[88] Vikas Dwivedi and Balaji Srinivasan. Physics informed extreme learning machine (PIELM)–a rapid method for the numerical solution of partial differential equations. *Neurocomputing*, 391:96–118, 2020.

[89] Ameya D Jagtap, Ehsan Kharazmi, and George Em Karniadakis. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365:113028, 2020.

[90] Xuhui Meng, Zhen Li, Dongkun Zhang, and George Em Karniadakis. PPINN: parareal physics-informed neural network for time-dependent PDEs. *Computer Methods in Applied Mechanics and Engineering*, 370:113250, 2020.

[91] Enrico Schiassi, Roberto Furfaro, Carl Leake, Mario De Florio, Hunter Johnston, and Daniele Mortari. Extreme theory of functional connections: A fast physics-informed neural network method for solving ordinary and partial differential equations. *Neurocomputing*, 457:334–356, 2021.

[92] Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021.

[93] Yunlei Yang, Muzhou Hou, and Jianshu Luo. A novel improved extreme learning machine algorithm in solving ordinary differential equations by Legendre neural network methods. *Advances in Difference Equations*, 2018(1):1–24, 2018.

[94] Sifan Wang, Xinling Yu, and Paris Perdikaris. When and why pinns fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449:110768, 2022.

[95] Evangelos Galaris, Gianluca Fabiani, Ioannis Gallos, Ioannis Kevrekidis, and Constantinos Siettos. Constructing coarse-scale bifurcation diagrams from spatio-temporal observations of microscopic simulations: A parsimonious machine learning approach, 2022.

[96] Dirk Husmeier. Random vector functional link (RVFL) networks. In *Neural Networks for Conditional Probability Estimation*, pages 87–97. Springer, 1999.

[97] Boris Igelnik and Yoh-Han Pao. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. *IEEE Transactions on Neural Networks*, 6(6):1320–1329, 1995.

[98] Yoh-Han Pao, Gwang-Hoon Park, and Dejan J. Sobajic. Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing*, 6(2):163–180, 1994.

[99] W.F. Schmidt, M.A. Kraaijveld, and R.P.W. Duin. Feedforward neural networks with random weights. In *Proceedings 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems*, pages 1–4. IEEE Computer Society Press, 1992.

[100] Herbert Jaeger. Adaptive nonlinear system identification with echo state networks. *Advances in Neural Information Processing Systems*, 15:609–616, 2002.

[101] Mustafa C. Ozturk, Dongming Xu, and Jose C. Principe. Analysis and design of echo state networks. *Neural Computation*, 19(1):111–138, 2007.

[102] Yusuke Sakemi, Kai Morino, Timothée Leleu, and Kazuyuki Aihara. Model-size reduction for reservoir computing by concatenating internal states through time. *Scientific Reports*, 10(1):1–13, 2020.

[103] G. Huang, H. Zhou, X. Ding, and R. Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):513–529, 2012.

[104] Guang-Bin Huang. An insight into extreme learning machines: random neurons, random features and kernels. *Cognitive Computation*, 6(3):376–390, 2014.

[105] Guang-Bin Huang. What are extreme learning machines? Filling the gap between Frank Rosenblatt's dream and John von Neumann's puzzle. *Cognitive Computation*, 7(3):263–278, 2015.

[106] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26(1):189–206, 1984.

[107] Guang-Bin Huang, Lei Chen, Chee Kheong Siew, et al. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Networks*, 17(4):879–892, 2006.

[108] A Prothero and A Robinson. On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations. *Mathematics of Computation*, 28(125):145–162, 1974.

[109] Balth Van der Pol. Lxxxviii. on "relaxation-oscillations". *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):978–992, 1926.

[110] H.H. Robertson. The solution of a set of reaction rate equations. *Numerical Analysis: an Introduction*, 178182, 1966.

[111] Evangelos Galaris, Ioannis Gallos, Ivan Myatchin, Lieven Lagae, and Constantinos Siettos. Electroencephalography source localization analysis in epileptic children during a visual working-memory task. *International Journal for Numerical Methods in Biomedical Engineering*, 36(12):e3404, 2020.

[112] Roberto D. Pascual-Marqui. Standardized low-resolution brain electromagnetic tomography (sloreta): technical details. *Methods and findings in experimental and clinical pharmacology*, 24D:5–12, 2002.

[113] Anders M. Dale, Arthur K. Liu, Bruce R. Fischl, Randy L. Buckner, John W. Belliveau, Jeffrey D. Lewine, and Eric Halgren. Dynamic statistical parametric mapping: Combining fmri and meg for high-resolution imaging of cortical activity. *Neuron*, 26(1):55 – 67, 2000.

[114] Sunao Iwaki and Shoogo Ueno. Weighted minimum-norm source estimation of magnetoencephalography utilizing the temporal information of the measured data. *Journal of Applied Physics*, 83:6441–6443, 06 1998.

[115] Yuki Otsuka, Naoyuki Osaka, and Mariko Osaka. Functional asymmetry of superior parietal lobule for working memory in the elderly. *Neuroreport*, 19:1355–9, 10 2008.

[116] Michael Koenigs, Aron K. Barbey, Bradley R. Postle, and Jordan Grafman. Superior parietal cortex is critical for the manipulation of information in working memory. *Journal of Neuroscience*, 29(47):14980–14986, 2009.

[117] Ingrid R. Olson and Marian Berryhill. Some surprising findings on the involvement of the parietal lobe in human memory. *Neurobiology of Learning and Memory*, 91(2):155 – 165, 2009. Special Issue: Parietal Cortex.

[118] Susan Ravizza, Marlene Behrmann, and Julie Fiez. Right parietal contributions to verbal working memory: Spatial or executive? *Neuropsychologia*, 43:2057–67, 02 2005.

[119] Manfred Fuchs, Michael Wagner, and Jörn Kastner. Boundary element method volume conductor models for eeg source reconstruction. *Clinical Neurophysiology*, 112(8):1400 – 1407, 2001.

[120] Carsten H. Wolters. The finite element method in eeg/meg source analysis. 2007.

[121] Hans Hallez, Bart Vanrumste, Roberta Grech, Joseph Muscat, Wim De Clercq, Anneleen Vergult, Yves D'Asseler, Kenneth P. Camilleri, Simon G. Fabri, Sabine Van Huffel, and Ignace Lemahieu. Review on solving the forward problem in eeg source analysis. *Journal of NeuroEngineering and Rehabilitation*, 4(1):46, Nov 2007.

[122] Roberto D Pascual-Marqui and Rolando Jose Lirio. Spatial resolution of neuronal generators based on eeg and meg measurements. *The International journal of neuroscience*, 68:93–105, 02 1993.

[123] A. N. Tikhonov and Vasiliy Yakovlevich Arsenin. Solutions of ill-posed problems. 1977.

[124] A. N. Tikhonov, Alexander Goncharsky, Vladimir V. Stepanov, and Anatoly G. Yagola. Numerical methods for the solution of ill-posed problems. 1995.

[125] M. S. Hämäläinen and R. J. Ilmoniemi. Interpreting magnetic fields of the brain: minimum norm estimates. *Medical & Biological Engineering & Computing*, 32(1):35–42, Jan 1994.

[126] Matti Stenroos and Olaf Hauk. Minimum-norm cortical source estimation in layered head models is robust against skull conductivity error. *NeuroImage*, 81:265 – 272, 2013.

[127] George Backus and Freeman Gilbert. The Resolving Power of Gross Earth Data. *Geophysical Journal International*, 16(2):169–205, 10 1968.

[128] H. L. Swanson. Cross-sectional and incremental changes in working memory and mathematical problem solving. *Journal of Educational Psychology*, 98(2):265 – 281, 2006.

[129] Susan E. Gathercole, Susan J. Pickering, Camilla Knight, and Zoe Stegmann. Working memory skills and educational attainment: evidence from national curriculum assessments at 7 and 14 years of age. *Applied Cognitive Psychology*, 18(1):1–16, 2004.

[130] Susan E. Gathercole, Tracy P. Alloway, Hannah J. Kirkwood, Julian G. Elliott, Joni Holmes, and Kerry A. Hilton. Attentional and executive function behaviours in children with poor working memory. *Learning and Individual Differences*, 18(2):214 – 223, 2008.

[131] Jozsef A. Toth and Michael Lewis. The role of working memory and external representation in individual decision making. *AAAI Technical Report*, 11, 05 2003.

[132] Christian C. Ruff, Markus Knauff, Thomas Fangmeier, and Joachim Spreer. Reasoning and working memory: common and distinct neuronal processes. *Neuropsychologia*, 41(9):1241 – 1253, 2003.

[133] Sibel Gülgönen, Veysi Demirbilek, Bariş Korkmaz, Ayşm Dervent, and Brenda D. Townes. Neuropsychological functions in idiopathic occipital lobe epilepsy. *Epilepsia*, 41(4):405–411, 2000.

[134] Yves Chaix, Virginie Laguitton, Valérie Lauwers-Cancès, Géraldine Daquin, Claude Cancès, Jean-François Démonet, and Nathalie Villeneuve. Reading abilities and cognitive functions of children with epilepsy: Influence of epileptic syndrome. *Brain and Development*, 28(2):122 – 130, 2006.

[135] Pablo Campo, Fernando Maestú, Irene García-Morales, Antonio Gil-Nagel, Bryan Strange, Manuel Morales, and Tomás Ortiz. Modulation of medial temporal lobe activity in epilepsy patients with hippocampal sclerosis during verbal working memory. *Journal of the International Neuropsychological Society : JINS*, 15:536–46, 08 2009.

[136] Dylan D. Wagner, Viviane Sziklas, Krista E. Garver, and Marilyn Jones-Gotman. Material-specific lateralization of working memory in the medial temporal lobe. *Neuropsychologia*, 47(1):112 – 122, 2009.

[137] Lindsay M. Luton, Thomas G. Burns, and Nick DeFilippis. Frontal Lobe Epilepsy in Children and Adolescents: A Preliminary Neuropsychological Assessment of Executive Function. *Archives of Clinical Neuropsychology*, 25(8):762–770, 09 2010.

[138] Arnaud Delorme and Scott Makeig. Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis. *Journal of Neuroscience Methods*, 134(1):9 – 21, 2004.

[139] John E. Richards, Carmen Sanchez, Michelle Phillips-Meek, and Wanze Xie. A database of age-appropriate average mri templates. *NeuroImage*, 124:1254–1259, 2016.

[140] Alan C. Evans. The nih mri study of normal brain development. *NeuroImage*, 30(1):184 – 202, 2006.

[141] Carmen E. Sanchez, John E. Richards, and C. Robert Almli. Age-specific mri templates for pediatric neuroimaging. *Developmental Neuropsychology*, 37(5):379–399, 2012. PMID: 22799759.

[142] John E. Richards and Wanze Xie. Chapter one - brains for all the ages: Structural neurodevelopment in infants and children from a life-span perspective. volume 48 of *Advances in Child Development and Behavior*, pages 1 – 52. JAI, 2015.

[143] Katrina Wendel-Mitoraj, Juho Väisänen, Gunnar Seemann, Jari Hyttinen, and Jaakko Malmivuo. The influence of age and skull conductivity on surface and subdermal bipolar eeg leads. *Computational Intelligence and Neuroscience*, 2010:397272, 01 2010.

[144] Hannah McCann, Giampaolo Pisano, and Leandro Beltrachini. Variation in reported human head tissue electrical conductivity values. *Brain Topography*, 32(5):825–858, Sep 2019.

[145] Sylvain Baillet, John Mosher, R.M. Leahy, and D.W. Shattuck. Brainstorm: A matlab toolbox for the processing of meg and eeg signals. *NeuroImage*, 9:S246, 01 1999.

[146] Alexandre Gramfort, Théodore Papadopoulo, Emmanuel Olivi, and Maureen Clerc. Openmeeg: Opensource software for quasistatic bioelectromagnetics. *Biomedical engineering online*, 9:45, 09 2010.

[147] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis Engemann, Daniel Strohmeier, Christian Brodbeck, Lauri Parkkonen, and Matti Hämäläinen. Mne software for processing meg and eeg data. *NeuroImage*, 86, 10 2013.

[148] Ahmed BenSaida. Shapiro-wilk and shapiro-francia normality tests. *MATLAB Central File Exchange. https://www.mathworks.com/matlabcentral/fileexchange/13964-shapiro-wilk-and-shapiro-francia-normality-tests)*, December 2019.

[149] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate - a practical and powerful approach to multiple testing. *J. Royal Statist. Soc., Series B*, 57:289 – 300, 11 1995.

[150] Aleksandar Ristić, Andreas Alexopoulos, Norman So, Chong Wong, and Imad Najm. Parietal lobe epilepsy: The great imitator among focal epilepsies. *Epileptic disorders : international epilepsy journal with videotape*, 14:22–31, 03 2012.

[151] Fabrice Bartolomei, Martine Gavaret, Russell Hewett, Luc Valton, Sandrine Aubert, Jean Régis, Fabrice Wendling, and Patrick Chauvel. Neural networks underlying parietal lobe seizures: A quantified study from intracerebral recordings. *Epilepsy Research*, 93(2):164 – 176, 2011.

[152] Juliana V. Baldo and Nina F. Dronkers. The role of inferior parietal and inferior frontal cortex in working memory. *Neuropsychology*, 20 5:529–38, 2006.

[153] Susan M. Ravizza, Marlene Behrmann, and Julie A. Fiez. Right parietal contributions to verbal working memory: Spatial or executive? *Neuropsychologia*, 43(14):2057 – 2067, 2005.

[154] Ingrid R. Olson and Marian Berryhill. Some surprising findings on the involvement of the parietal lobe in human memory. *Neurobiology of Learning and Memory*, 91(2):155 – 165, 2009. Special Issue: Parietal Cortex.

[155] Marian E. Berryhill and Ingrid R. Olson. Is the posterior parietal lobe involved in working memory retrieval?: Evidence from patients with bilateral parietal lobe damage. *Neuropsychologia*, 46(7):1775 – 1786, 2008. Part Special Issue: What is the Parietal Lobe Contribution to Human Memory?

[156] Kevin Jones and Marian Berryhill. Parietal contributions to visual working memory depend on task difficulty. *Frontiers in Psychiatry*, 3:81, 2012.

[157] Mariko Osaka, Mie Komori, Masanao Morishita, and Naoyuki Osaka. Neural bases of focusing attention in working memory: An fmri study based on group differences. *Cognitive, affective and behavioral neuroscience*, 7:130–9, 07 2007.

[158] Song Li, Jing-Na Jin, Xin Wang, Hong-Zhi Qi, Zhi-Peng Liu, and Tao Yin. Theta and alpha oscillations during the retention period of working memory by rtms stimulating the parietal lobe. *Frontiers in Behavioral Neuroscience*, 11:170, 2017.

[159] Juan CH Tseng P, Iu KC. The critical role of phase difference in theta oscillation between bilateral parietal cortices for visuospatial working memory. *Scientific reports*, 8(1):349, 01 2018.

[160] Martin Seeber, Lucia Cantonas, Mauritius Hoevels, Thibaut Sesia, Veerle Visser-Vandewalle, and Christoph Michel. Subcortical electrophysiological activity is detectable with high-density eeg source imaging. *Nature Communications*, 10, 12 2019.

[161] Jasmine Song, Colin Davey, Catherine Poulsen, Phan Luu, Sergei Turovets, Erik Anderson, Kai Li, and Don Tucker. Eeg source localization: Sensor density and head surface coverage. *Journal of Neuroscience Methods*, 256:9 – 21, 2015.

[162] G Lantz, R Grave de Peralta, L Spinelli, M Seeck, and C.M Michel. Epileptic source localization with high density eeg: how many electrodes are needed? *Clinical Neurophysiology*, 114(1):63 – 69, 2003.

[163] F. Dan Foresee and M.T. Hagan. Gauss-newton approximation to bayesian learning. In *Proceedings of International Conference on Neural Networks (ICNN'97)*, volume 3, pages 1930–1935 vol.3, 1997.

[164] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

[165] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[166] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, 3(5):551–560, 1990.

[167] Jooyoung Park and Irwin W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 3(2):246–257, 1991.

[168] Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

[169] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

[170] Daniela di Serafino, Nataša Krejić, Nataša Krklec Jerinkić, and Marco Viola. LSOS: line-search second-order stochastic optimization methods for nonconvex finite sums. *arXiv preprint arXiv:2007.15966*, 2021.

[171] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

[172] Santosh S. Vempala. *The random projection method*, volume 65. American Mathematical Soc., 2005.

[173] B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302 – 1338, 2000.

[174] John B Butcher, David Verstraeten, Benjamin Schrauwen, Charles R Day, and Peter W Haycock. Reservoir computing and extreme learning machines for non-linear time-series data analysis. *Neural Networks*, 38:76–89, 2013.

[175] Raja Giryes, Guillermo Sapiro, and Alex M Bronstein. Deep neural networks with random Gaussian weights: a universal classification strategy? *IEEE Transactions on Signal Processing*, 64(13):3444–3457, 2016.

[176] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

[177] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

[178] Weipeng Cao, Xizhao Wang, Zhong Ming, and Jinzhu Gao. A review on neural networks with random weights. *Neurocomputing*, 275:278–287, 2018.

[179] Siyuan Gao, Gal Mishne, and Dustin Scheinost. Nonlinear manifold learning in functional magnetic resonance imaging uncovers a low-dimensional space of brain dynamics. *Human Brain Mapping*, 42, 06 2021.

[180] C.J. Dsilva, R. Talmon, R.R. Coifman, and I.G. Kevrekidis. Parsimonious representation of nonlinear dynamical systems through manifold learning: A chemotaxis case study. *Applied and Computational Harmonic Analysis*, 44(3):759–773, 2018.

[181] Alexander Holiday, Mahdi Kooshkbaghi, Juan M. Bello-Rivas, C. William Gear, Antonios Zagaris, and Ioannis G. Kevrekidis. Manifold learning for parameter reduction. *Journal of Computational Physics*, 392:419–431, 2019.

[182] Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. DeepXDE: a deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021.

[183] Carl T. Kelley. *Iterative methods for optimization*. SIAM, 1999.

[184] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[185] Lawrence F. Shampine and Mark W. Reichelt. The MATLAB ODE suite. *SIAM Journal on Scientific Computing*, 18(1):1–22, 1997.

[186] Fadil Santosa and William W. Symes. Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1307–1330, 1986.

[187] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[188] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1, 1995.

[189] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[190] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[191] Richard FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6):445–466, 1961.

[192] Constantinos Theodoropoulos, Yue-Hong Qian, and Ioannis G. Kevrekidis. "coarse" stability and bifurcation analysis using time-steppers: A reaction-diffusion example. *Proceedings of the National Academy of Sciences*, 97(18):9840–9843, 2000.

[193] Shiyi Chen and Gary D. Doolen. Lattice boltzmann method for fluid flows. *Annual Review of Fluid Mechanics*, 30(1):329–364, 1998.

[194] Julia M. Yeomans. The lattice boltzmann equation for fluid dynamics and beyond. *Physics Today*, 55(12):58–, 60, 2002.

[195] P. L. Bhatnagar, E. P. Gross, and M. Krook. A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems. *Phys. Rev.*, 94:511–525, May 1954.

[196] P Bhathnagor, E Gross, and Max Krook. A model for collision processes in gases. *Physical Review*, 94(3):511, 1954.

[197] Y.H. Qian and S.A. Orszag. Scalings in diffusion-driven reaction $a+b \rightarrow c$: Numerical simulations by lattice bgk models. *J Stat Phys*, 81:237–253, 1995.

[198] Tony F. C. Chan and H. B. Keller. Arc-length continuation and multigrid techniques for nonlinear elliptic eigenvalue problems. *SIAM Journal on Scientific and Statistical Computing*, 3(2):173–194, 1982.

[199] Roland Glowinski, Herbert B. Keller, and Laure Reinhart. Continuation-conjugate gradient methods for the least squares solution of nonlinear boundary value problems. *Siam Journal on Scientific and Statistical Computing*, 6:793–832, 1985.

[200] Willy Govaerts. Numerical methods for bifurcations of dynamical equilibria. 1987.

[201] Daniel Lehmberg, Felix Dietrich, Gerta Köster, and Hans-Joachim Bungartz. datafold: data-driven models for point clouds and time series on manifolds. *Journal of Open Source Software*, 5:2283, 07 2020.

[202] Robert J. Lovelett, José L. Avalos, and Ioannis G. Kevrekidis. Partial observations and conservation laws: Gray-box modeling in biotechnology and optogenetics. *Industrial & Engineering Chemistry Research*, 59(6):2611–2620, 2020.

[203] Or Yair, Ronen Talmon, Ronald R. Coifman, and Ioannis G. Kevrekidis. Reconstruction of normal forms by learning informed observation geometries from data. *Proceedings of the National Academy of Sciences*, 114(38):E7865–E7874, 2017.

[204] J.C. Chen, Kung Yao, and R.E. Hudson. Source localization and beamforming. *IEEE Signal Processing Magazine*, 19(2):30–39, 2002.

[205] Paolo Chiariotti. Acoustic beamforming for noise source localization – reviews, methodology and applications. *Mechanical Systems and Signal Processing*, 120:422–448, 04 2019.

[206] Dimitrios Pantazis and Amir Adler. Meg source localization via deep learning, 2020.

[207] Chen Wei, Kexin Lou, Zhengyang Wang, Mingqi Zhao, Dante Mantini, and Quanying Liu. Edge sparse basis network: A deep learning framework for eeg source localization, 2021.

[208] Harshit Gupta, Kyong Hwan Jin, Ha Q. Nguyen, Michael T. McCann, and Michael Unser. Cnn-based projected gradient descent for consistent ct image reconstruction. *IEEE Transactions on Medical Imaging*, 37(6):1440–1453, Jun 2018.

[209] Kyong Hwan Jin, Michael T. McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.

[210] Dong Liang, Jing Cheng, Ziwen Ke, and Leslie Ying. Deep magnetic resonance image reconstruction: Inverse problems meet neural networks. *IEEE Signal Processing Magazine*, 37(1):141–151, 2020.

[211] Jo Schlemper, Jose Caballero, Joseph V. Hajnal, Anthony Price, and Daniel Rueckert. A deep cascade of convolutional neural networks for dynamic mr image reconstruction, 2017.

[212] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks, 2015.

[213] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks, 2016.

[214] Mauricio Araya-Polo, Joseph Jennings, Amir Adler, and Taylor Dahlke. Deep-learning tomography. *The Leading Edge*, 37(1):58–66, 2018.

[215] M. Hassan and F. Wendling. Electroencephalography source connectivity: Aiming for high resolution of brain networks in time and space. *IEEE Signal Processing Magazine*, 35(3):81–96, May 2018.

[216] Mahmoud Hassan, Olivier Dufor, Isabelle Merlet, Claude Berrou, and Fabrice Wendling. Eeg source connectivity analysis: From dense array recordings to brain networks. *PLOS ONE*, 9(8):1–15, 08 2014.

[217] Elham Barzegaran and Maria G. Knyazeva. Functional connectivity analysis in eeg source space: The choice of method. *PLOS ONE*, 12(7):1–16, 07 2017.

[218] Bahar Moezzi and Mitchell R. Goldsworthy. Commentary: Consistency of eeg source localization and connectivity estimates. *Frontiers in Neuroscience*, 12:147, 2018.

[219] Willeke Staljanssens, Gregor Strobbe, Roel Van Holen, Vincent Keereman, Stefanie Gadeyne, Evelien Carrette, Alfred Meurs, Francesca Pittau, Shahan Momjian, Margitta Seeck, Paul Boon, Stefaan Vandenberghe, Serge Vulliemoz, Kristl Vonck, and Pieter van Mierlo. Eeg source connectivity to localize the seizure onset zone in patients with drug resistant epilepsy. *NeuroImage: Clinical*, 16:689 – 698, 2017.

[220] Lin Gao, Linda Sommerlade, Brian Coffman, Tongsheng Zhang, Julia Stephen, Dichen Li, Jue Wang, Celso Grebogi, and Bjoern Schelter. Granger causal time-dependent source connectivity in the somatosensory network. *Scientific Reports*, 5:10399, 05 2015.

[221] Luiz Baccala and Koichi Sameshima. Partial directed coherence: A new concept in neural structure determination. *Biological Cybernetics*, 84:463–474, 05 2001.

[222] Lukas Meier, Sara Van De Geer, and Peter Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71, 2008.

[223] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1, 1995.

[224] Xiangyan Zeng, Yen-Wei Chen, and Caixia Tao. Feature selection using recursive feature elimination for handwritten digit recognition. In *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 1205–1208, 2009.

[225] Shivani Gupta and Atul Gupta. Dealing with noise problem in machine learning data-sets: A systematic review. *Procedia Computer Science*, 161:466–474, 2019. The Fifth Information Systems International Conference, 23-24 July 2019, Surabaya, Indonesia.

[226] Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed deeponets. *Science Advances*, 7(40):eabi8605, 2021.

[227] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces. *arXiv preprint arXiv:2108.08481*, 2021.

[228] Walter M Lioen and Jacques JB de Swart. *Test set for initial value problem solvers*. Centrum voor Wiskunde en Informatica, 1998.

[229] Francesca Mazzia, Jeff R Cash, and Karline Soetaert. A test set for stiff initial value problem solvers in the open source software R: Package detestset. *Journal of Computational and Applied Mathematics*, 236(16):4119–4131, 2012.

[230] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Numerical Gaussian processes for time-dependent and nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 40(1):A172–A198, 2018.

[231] Jin-Long Wu, Heng Xiao, and Eric Paterson. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Physical Review Fluids*, 3(7):074602, 2018.